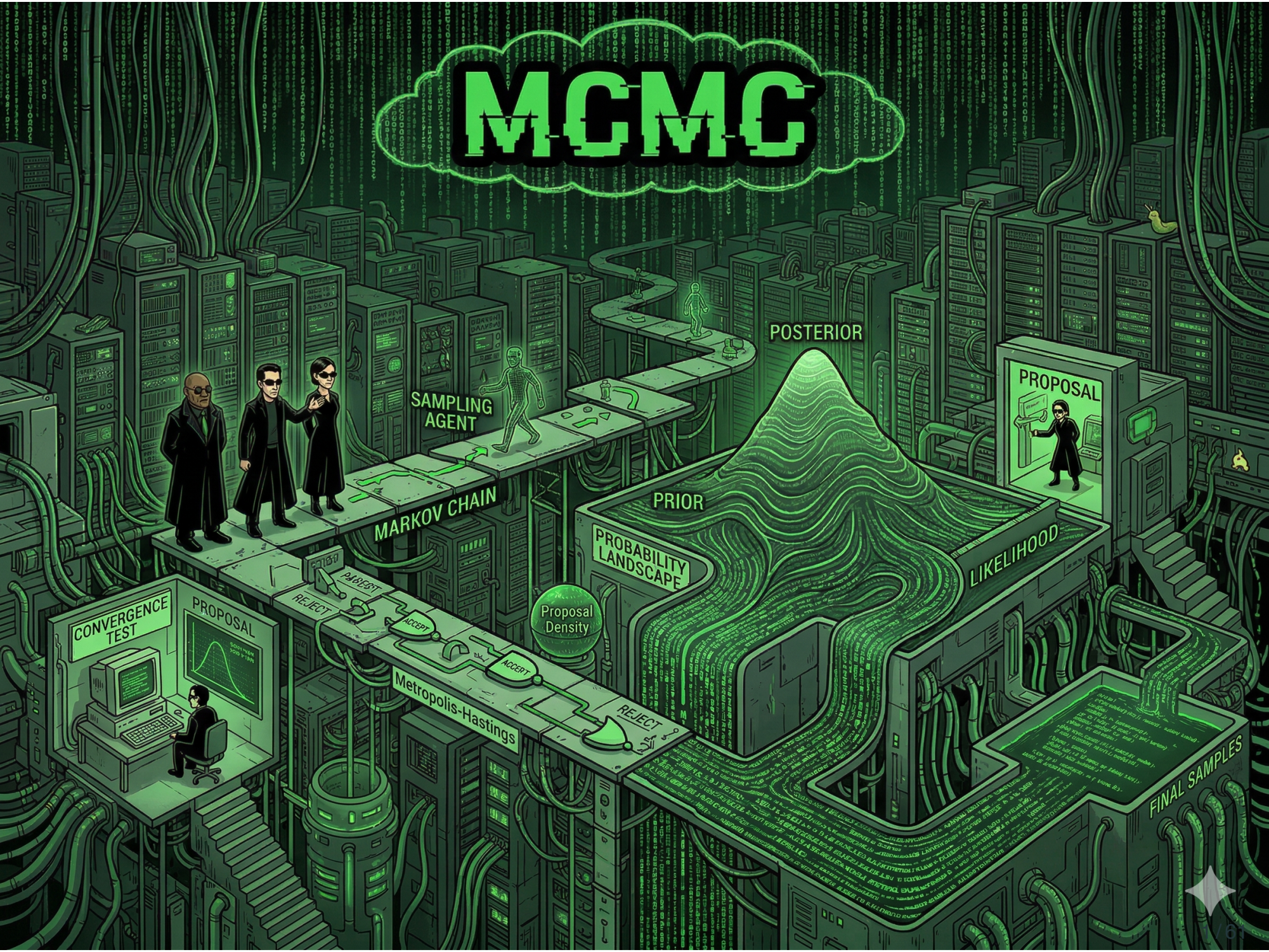

MCMC
SAMPLING AGENT
MARKOV CHAIN
POSTERIOR
PRIOR
PROBABILITY LANDSCAPE
LIKELIHOOD
PROPOSAL
CONVERGENCE TEST
PROPOSAL
REJECT
ACCEPT
Proposal Density
ACCEPT
REJECT
Metropolis-Hastings
FINAL SAMPLES


Markov Chain Monte Carlo
samplijing
probability
landscape
sampling
imputa
spirits

Overview

- This lecture introduces the theoretical foundations of MCMC and cover some practical considerations
  - Markov chain fundamentals: Definition and properties
  - Ergodic theorem: Convergence to stationary distribution
  - Metropolis-Hastings: A general MCMC algorithm
  - Gibbs sampling: A special case with automatic acceptance
  - Combining moves: Sequentially or randomly
  - MCMC diagnostics: Assessing convergence

MCMC: review

- $\{X_t\}$ is a Markov chain if for all t, $X_{t+1} \perp (X_1, ..., X_{t-1})|X_t$. That is, $p(x_{t+1}|x_{1:t}) = p(x_{t+1}|x_t)$ for all $x_{1:t+1}$.
- Let us focus on discrete X here. The same results extend to continuous case.
- We say $\{X_t\}$ is time-homogeneous if the distribution of $X_{t+1}|X_t$ is the same for all t, i.e., $p(X_{t+1} = b|X_t = a) = T_{ab}$. T is called the transition matrix.
- π is a stationary distribution for transition matrix T if for all b,

$$\sum_a \pi(a)T_{ab} = \pi(b)$$

  These system of linear equations is called the global balance equations.
- $\{X_t\}$ is irreducible if for all a, b, there is some t such that $p(X_t = b|X_0 = a) > 0$.
- $\{X_t\}$ is aperiodic if for all a, $gcd(\{t : p(X_t = a|X_0 = a) > 0\}) = 1$, where gcd denotes the greatest common divisor. That is, the times we return to the same state are not periodic.

MCMC: review

Theorem (Ergodic theorem)

If $\{X_t\}$ is a time-homogeneous, irreducible, discrete Markov chain with stationary distribution π, then for any function $h(x)$ such that $\mathbb{E}_\pi|h(x)| < \infty$,

$$\frac{1}{T}\sum_{t=1}^{T} h(X_t) \to \mathbb{E}_\pi(h(x))$$

as $T \to \infty$ almost surely (i.e., with probability 1). If further that $\{X_t\}$ is aperiodic, then for all x, x_0,

$$p(X_t = x | X_0 = x_0) \to \pi(x)$$

The general MCMC strategy is to construct an ergodic Markov chain $\{X_t\}$ with stationary distribution π that is the target distribution.

- Metropolis-Hastings Algorithm
- Gibbs sampling
- ...

MCMC: when stationary distribution exists?

- Let $t_a = \inf\{n \geqslant 1 : X_n = a\}$ be the return time to state a, then state a is called recurrent if $p(t_a < \infty | X_0 = a) = 1$, i.e., the chain returns to state a with probability 1. Otherwise, it is called transient.
  - State a is recurrent is equivalent to $\sum_{n=1}^{\infty} P(X_n = a | X_0 = a) = \infty$.
- It is called positive recurrent if $\mathbb{E}(t_a | X_0 = a) < \infty$.
- An irreducible Markov chain is recurrent if all states are recurrent. A recurrent chain is positive recurrent if all states are positive recurrent.
- Existence: If a Markov chain has at least one positive recurrent state, then there exists a stationary distribution π.
  - Example of a MC without positive recurrent state: $p(x_{i+1} = a + 1 | x_i = a) = 1$.
- Uniqueness: If a Markov chain is irreducible, there exists a stationary distribution and the distribution is unique if and only if the Markov chain is positive recurrent.

Examples of discrete Markov chains

Gambler's ruin

- A gambler plays a game where he can increase or decrease his fortune by 1 with probability p and $1-p$. The game stops at his fortune being 0 or 3. Then for any $a \in (0,1)$, the vector $\pi = (a, 0, 0, 1-a)$ satisfies the global balance condition and thus is a stationary distribution.
- The two absorbing states 0 and 3 are positive recurrent.
- Consider state 1, $P(T_1 < \infty | X_0 = 1) = 1 - P(T_1 = \infty | X_0 = 1) < 1$ thus states 1 and 2 are not recurrent.

1d random walk

- Let X_n be a random walk on the set of all integers $\mathbb{Z}$ with probability p of increasing by 1 and $1-p$ of decreasing by 1.
- This Markov chain is periodic.
- It turns out $p(X_{2n} = a | X_0 = a) \to (4p(1-p))^n / \sqrt{\pi n}$. Since $4p(1-p) \leqslant 1$, in order for the Markov chain to be recurrent, $\sum_n p(X_{2n} = a | X_0 = a) = \infty$, we need $p = 0.5$, i.e., the symmetric 1d random walk is recurrent (but not positive recurrent).
- Interestingly, the symmetric 2d random walk is also recurrent, but they are not recurrent for dimensions larger than 2.

Metropolis-Hasting: review

- A very useful property for studying MCMC is detailed balance. We say detailed balance holds if for all a, b, $\pi(a)T_{ab} = \pi(b)T_{ba}$.
- Detailed balance implies global balance, i.e., π is a stationary distribution for T.
- Suppose the target distribution is $\pi(\theta)$. From each θ, we pick a proposal distribution $q(\theta^\star|\theta)$. Then the acceptance ratio is

$$\alpha(\theta, \theta^\star) = \frac{\pi(\theta^\star)q(\theta|\theta^\star)}{\pi(\theta)q(\theta^\star|\theta)}$$

- The M-H algorithm proceeds with
  1. Sample $\theta \sim q(\theta|\theta^{(t-1)})$.
  2. Sample $u \in Unif(0, 1)$. If $u < \alpha(\theta^{(t-1)}, \theta)$ then set $\theta^{(t)} = \theta$, otherwise, set $\theta^{(t)} = \theta^{(t-1)}$.
- The M-H algorithm defines a Markov chain with transition matrix T s.t.

$$T_{ab} = q(b|a) \min\{1, \frac{\pi(b)q(a|b)}{\pi(a)q(b|a)}\}$$

  when $a \neq b$ and for all a, $T_{aa} = 1 - \sum_{b \neq a} T_{ab}$.
- We can verify that $\pi(a)T_{ab} = \pi(b)T_{ba}$.

Gibbs sampling: review

- Gibbs sampling is a special case of M-H algorithm.
- Let $\pi(\theta, \phi)$ be the target distribution. At iteration t, the Gibbs update can be seen as sampling from

$$q(\theta, \phi | \theta^{(t-1)}, \phi^{(t-1)}) = \pi(\theta | \phi^{(t-1)}) I(\phi = \phi^{(t-1)})$$

- With probability 1, $\phi = \phi^{(t-1)}$. The acceptance probability then is always 1 since

$$\begin{aligned}
\alpha((\theta, \phi), (\theta^{(t-1)}, \phi^{(t-1)})) &= \frac{\pi(\theta, \phi) q(\theta^{(t-1)}, \phi^{(t-1)} | \theta, \phi)}{\pi(\theta^{(t-1)}, \phi^{(t-1)}) q(\theta, \phi \mid \theta^{(t-1)}, \phi^{(t-1)})} \\
&= \frac{\pi(\theta, \phi) \pi(\theta^{(t-1)} \mid \phi)}{\pi(\theta^{(t-1)}, \phi) \pi(\theta \mid \phi)} \\
&= \frac{\pi(\theta \mid \phi) \pi(\phi) \pi(\theta^{(t-1)} \mid \phi)}{\pi(\theta^{(t-1)} \mid \phi) \pi(\phi) \pi(\theta \mid \phi)} \\
&= 1
\end{aligned}$$

Combining MCMC moves

- A nice feature of MCMC is that it is easy to combine different moves when constructing a sampler.
- Roughly, a move is a way of updating the variables using an MCMC step targeting π, i.e., a transition matrix T such that $\pi T = \pi$.
- If $T_1, ..., T_k$ all have stationary distribution π, then the product $T = T_1 T_2 \cdots T_k$ also has stationary distribution π.
- That is, we can cycle through all variables in a deterministic order in each step of update.
- For example, the sequential scan or fixed scan Gibbs sampling algorithm:
  - Sample $\theta_1^t \sim \pi_1(\theta_1 | \theta_{2:p}^{(t-1)})$
  - Sample $\theta_2^t \sim \pi_2(\theta_2 | \theta_1^t, \theta_{3:p}^{(t-1)})$
  - ...
  - Sample $\theta_p^t \sim \pi_p(\theta_p | \theta_{1:(p-1)}^t)$
- Another example is the Metropolis-Hastings-within-Gibbs algorithm, which replaces some of the sub-steps with a M-H step.

Combining MCMC moves

- Another observation is that for any $\omega_1, ..., \omega_k \geqslant 0$ and $\sum \omega_i = 1$, $T = \sum_{i=1}^{k} \omega_i T_i$ also has stationary distribution π.
- This motivates the random scan Gibbs sampling algorithm, where at each step t,
  - Sample index i by drawing a random variable with probability mass function $\{\omega_1, ..., \omega_p\}$.
  - Sample $\theta_i^{(t)} \sim \pi_i(\theta_i | \theta_{-i}^{(t-1)})$
- Note that the random choice of move should not depend on the current state, otherwise it can fail to converge to the correct stationary distribution.

Example: MH within Gibbs

- We consider the estimation of the prevalence of Type II diabetes by age and sex in health reporting areas (HRAs) in King County, Washington.
- We use Behavioral Risk Factor Surveillance System (BRFSS) data.
- These survey data are collected using a complex stratified design, but we will ignore this aspect in this example.
- Consider the three-stage hierarchical model:

$$y_i|\theta_i \sim \text{Bin}(n_i, \theta_i)$$
$$\theta_i|a, b \sim \text{Beta}(a, b)$$
$$a, b \sim p(a, b)$$

for each of the i-th subpopulation defined by age, sex, and area.

Example: MH within Gibbs

- The prior for a and b does not have conjugate forms.
- One way to put a prior on (a, b) is by parameterization of the beta-binomial model. Marginally,

$$\mathbb{E}(Y_i) = n_i \mathbb{E}(\theta_i) = n_i \frac{a}{a+b}$$

$$\text{var}(Y_i) = n_i \mathbb{E}(\theta_i)(1 - \mathbb{E}(\theta_i))(1 + (n_i - 1)\frac{1}{a+b+1})$$

  where $\frac{1}{a+b+1}$ is usually referred to as the overdispersion parameter.
- A different parameterization of the beta-binomial distribution that is more commonly used in hierarchical modeling is via $\mu = \frac{a}{a+b}$, $d = \frac{1}{a+b+1}$. Different priors have been used in the literature. For illustration, here we let

$$p(a, b) \propto (a+b)^{-2.5}$$

  following the argument in Ch 5.3 of BDA3.

Example: MH within Gibbs

- We will use M-H steps to sample new values of a and b. We illustrate for a below.
- Since they are positive numbers, we do a random walk on the log scale. That is, we let

$$\phi = \log a^{new} = \log a^{old} + U^*$$

  where $U^* \in Unif(-\lambda_a/2, \lambda_a/2)$.
- Or equivalently, we generate new a by $a \exp(\lambda_a(U_1 - 0.5))$ where $U \sim Unif(0, 1)$ and λ_a is a tuning constant.
- Hence the proposal density is

$$q(a^{new}|a^{old}) = p(\phi|a^{old})|\frac{d\phi}{da^{new}}| = \frac{1}{\lambda_a a^{new}}$$

Beta-binomial details

Start with some initial values (θ, a, b). For $t = 0$ to T.

- For $i = 1$ to N, sample $\theta_i^{(t+1)} \sim$ Beta$(y_i + a^{(t)}, n_i - y_i + b^{(t)})$.
- Generate $U_1 \sim Unif(0, 1)$. Set $a^* = a^{(t)} \exp(\lambda_a(U_1 - 0.5))$. Accept $a^{(t+1)} = a^*$ with probability

$$\frac{\pi(\theta^{(1)}, a^*, b^{(t)})q(a^{(t)}|a^*)}{\pi(\theta^{(1)}, a^{(t)}, b^{(t)})q(a^*|a^{(t)})}$$

  otherwise, set $a^{(t+1)} = a^{(t)}$.
- Generate $U_2 \sim Unif(0, 1)$. Set $b^* = b^{(t)} \exp(\lambda_b(U_2 - 0.5))$. Accept $b^{(t+1)} = b^*$ with probability

$$\frac{\pi(\theta^{(1)}, a^{(t)}, b^*)q(b^{(t)}|b^*)}{\pi(\theta^{(1)}, a^{(t)}, b^{(t)})q(b^*|b^{(t)})}$$

  otherwise, set $b^{(t+1)} = b^{(t)}$.

Implementations in R

```
# log prior evaluation function
log.prior <- function(alpha, beta) {
  -2.5 * log(alpha + beta)
}
# sample proposal value
getstar = function(current, lambda) {
  return(current * exp(lambda * (runif(1) - 0.5)))
}
# conditional sample of p
draw.p <- function(alpha, beta, y, n) {
  return(rbeta(length(y), alpha + y, beta + n - y))
}
```

Implementations in R

```
draw.alpha <- function(alpha, beta, p, lambda) {
  size = length(p)
  alpha.star <- getstar(alpha, lambda)
  num <- size * (lgamma(alpha.star + beta) - lgamma(alpha.star)) +
  alpha.star * sum(log(p))
  num <- num + log.prior(alpha.star, beta)
  num <- num + log(alpha.star)
  den <- size * (lgamma(alpha + beta) - lgamma(alpha)) +
  alpha * sum(log(p))
  den <- den + log.prior(alpha, beta)
  den <- den + log(alpha)
  acc <- ifelse((log(runif(1)) <= num - den) && (alpha.star > 0), 1, 0)
  return(c(acc, ifelse(acc, alpha.star, alpha)))
}
```

Implementations in R

```
draw.beta <- function(alpha, beta, p, lambda) {
  size = length(p)
  beta.star <- getstar(beta, lambda)
  num <- size * (lgamma(alpha + beta.star) - lgamma(beta.star)) +
  beta.star * sum(log(1 - p))
  num <- num + log.prior(alpha, beta.star)
  num <- num + log(beta.star)
  den <- size * (lgamma(alpha + beta) - lgamma(beta)) +
  beta * sum(log(1 - p))
  den <- den + log.prior(alpha, beta)
  den <- den + log(beta)
  acc <- ifelse((log(runif(1)) <= num - den) && (beta.star > 0), 1, 0)
  return(c(acc, ifelse(acc, beta.star, beta)))
}
```

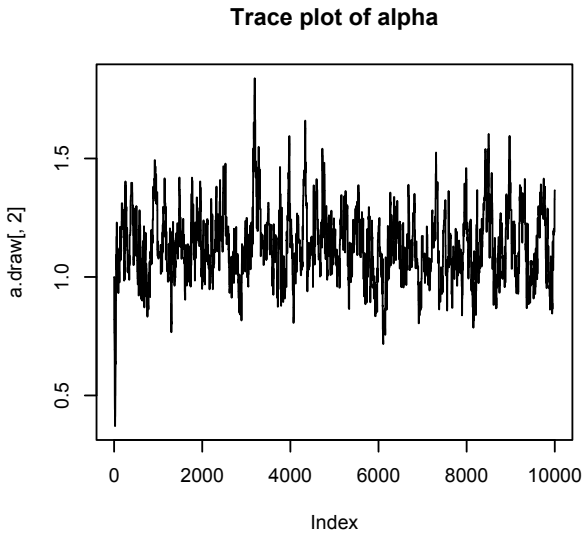
Implementations in R

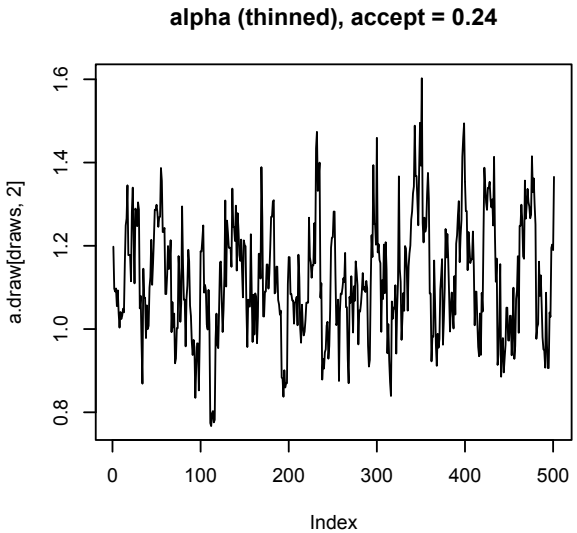
```
library(SUMMER)
library(dplyr)
data(BRFSS)
data <- BRFSS %>%
    group_by(hracode, sex, age4) %>%
    summarise(n = n(), y = sum(diab2)) %>%
    mutate(y = ifelse(is.na(y), 0, y))

Nitr <- 10000
a.draw <- b.draw <- matrix(0, Nitr, 2)
ps <- matrix(NA, nrow = Nitr, ncol = dim(data)[1])

# Metropolis tuning parameters
lambda.alpha <- .5
lambda.beta <- .5

# Initial values for the chain
a.draw[1, 2] <- 1
b.draw[1, 2] <- 1
ps[1, ] <- draw.p(a.draw[1, 2], b.draw[1, 2], y = data$y, n = data$n)
```

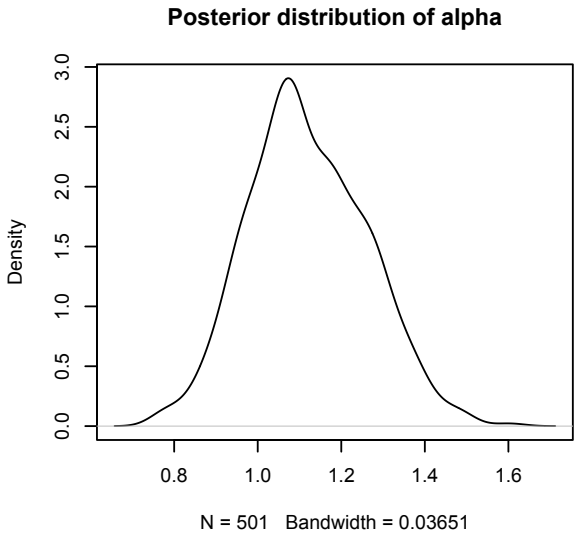
Implementations in R

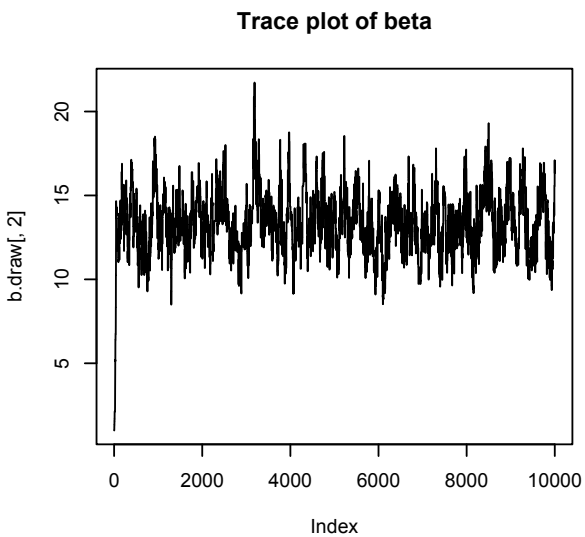
```
# MCMC simulation
for (m in 2:Nitr) {
  a.draw[m, ] <- draw.alpha(a.draw[m - 1, 2], b.draw[m -
  1, 2], ps[m - 1, ], lambda.alpha)
  b.draw[m, ] <- draw.beta(a.draw[m, 2], b.draw[m -
  1, 2], ps[m - 1, ], lambda.beta)
  ps[m, ] <- draw.p(a.draw[m, 2], b.draw[m, 2], y = data$y, n = data$n)
}

# thinning
draws <- seq(round(Nitr / 2), Nitr, by = 10)
par(mfcol = c(2, 3))
plot(a.draw[, 2], type = "l", main = "Trace plot of alpha")
plot(b.draw[, 2], type = "l", main = "Trace plot of beta")
plot(a.draw[draws, 2], type = "l", main = paste0("alpha (thinned),
     accept = ", round(mean(a.draw[draws, 1]), 2)))
plot(b.draw[draws, 2], type = "l", main = paste0("beta (thinned),
     accept = ", round(mean(b.draw[draws, 1]), 2)))
plot(density(a.draw[draws, 2]), main = "Posterior distribution of alpha")
plot(density(b.draw[draws, 2]), main = "Posterior distribution of beta")
```

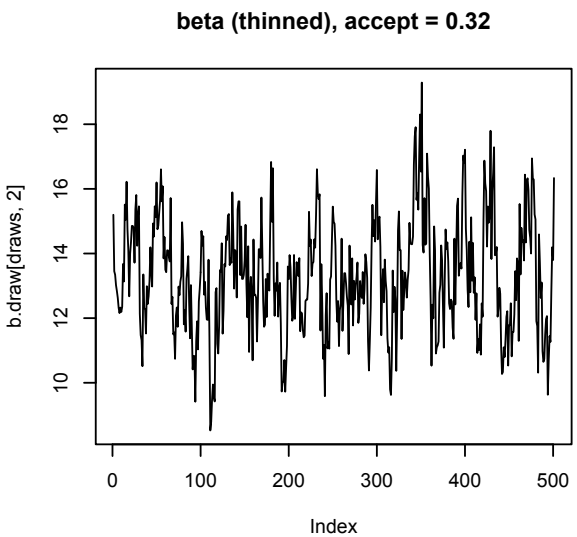
Results

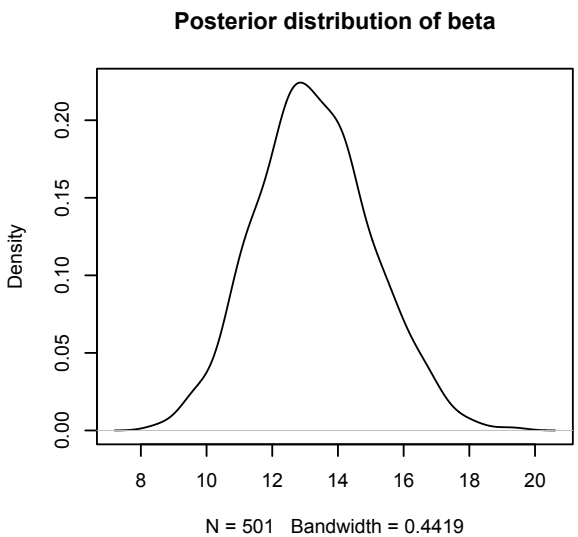

Trace plot of alpha
a.draw[, 2]
Index


alpha (thinned), accept = 0.24
a.draw[draws, 2]
Index

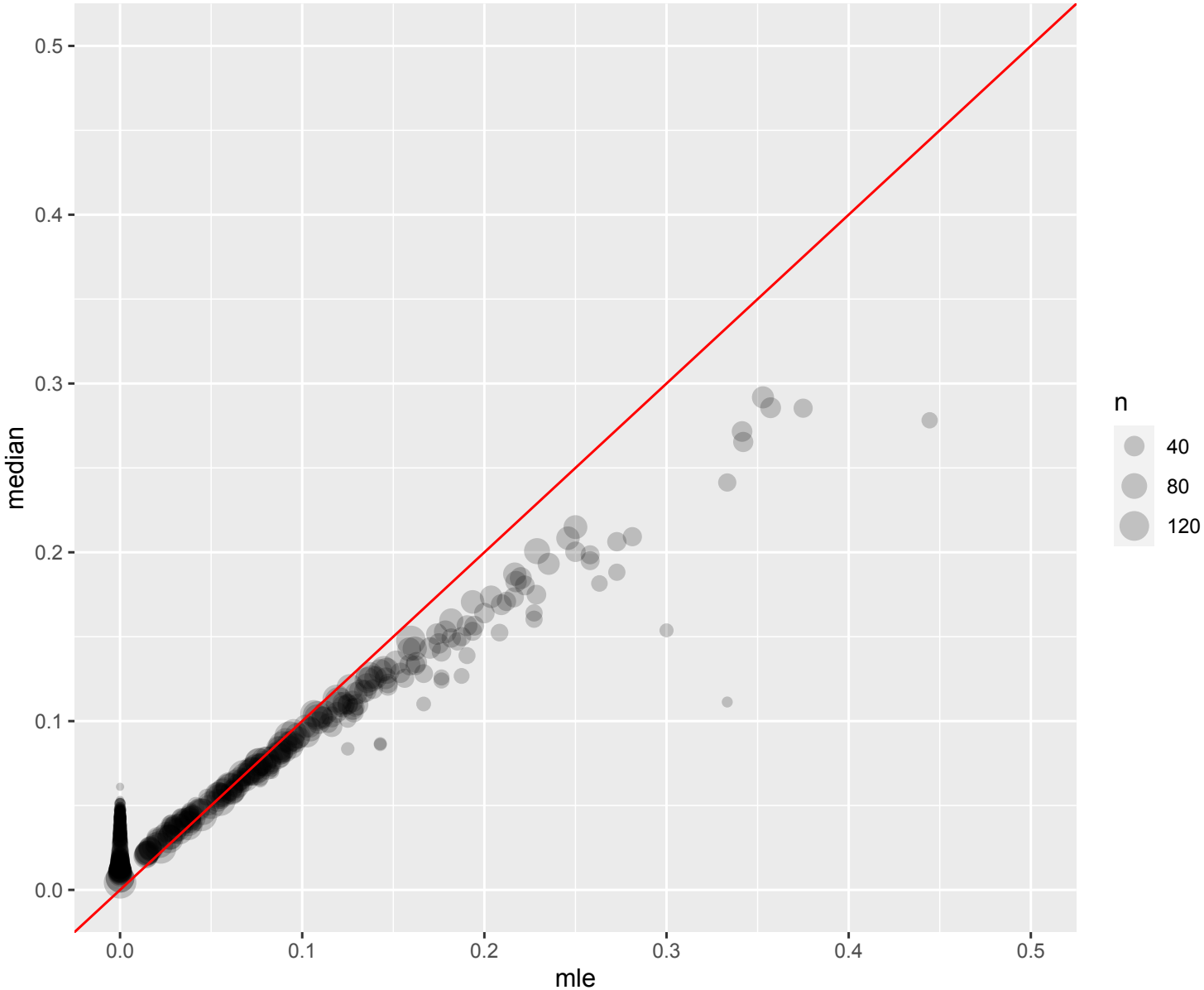

Posterior distribution of alpha
Density
N = 501 Bandwidth = 0.03651


Trace plot of beta
b.draw[, 2]
Index


beta (thinned), accept = 0.32
b.draw[draws, 2]
Index


Posterior distribution of beta
Density
N = 501 Bandwidth = 0.4419

Results



Reparameterize models

- The mixing of MCMC can sometimes be improved through reparameterization of the models, as the geometry of the parameter space will influence how fast the Markov chain explores the space.
- Consider the eight school example. Again let us assume σ_j is known for simplicity.

$$\begin{aligned} \bar{y}_{\cdot j}|\theta_j &\sim N(\theta_j, \sigma_j^2) \\ \theta_j|\mu, \tau^2 &\sim N(\mu, \tau^2) \\ p(\mu|\tau) \propto 1 \quad p(\tau) &\propto 1 \end{aligned}$$

- The naive sampler updates each value of θ_j first, and then update the values of μ and τ.
- When updating θ_j if τ is small then values of θ_j close to μ will have high conditional probability.
- Then when updating τ, small values will in return have high conditional probabilities too.
- The chain takes many steps to move from the space of small τ to large τ.

Eight schools: Gibbs sampler

We can derive the naive Gibbs sampler:

1. Sample $\theta_j|\mu,\tau,y \sim N(\theta_{0j}, V_j)$, where $\theta_{0j} = (\mu/\tau^2 + \bar{y}_{\cdot j}/\sigma_j^2)/(1/\tau^2 + 1/\sigma_j^2)$ and $V_j = 1/(1/\tau^2 + 1/\sigma_j^2)$.
2. Sample $\mu|\theta,\tau,y \sim N(\frac{1}{J}\sum\theta_j, \tau^2/J)$.
3. Sample $\tau^2 \sim$ Inv-$\chi^2(J-1, \frac{1}{J-1}\sum_j(\theta_j-\mu)^2)$ or Inv-Gamma$(\frac{J-1}{2}, \frac{\sum_j(\theta_j-\mu)^2}{2})$.

```
theta_update <- function(mu, tau, sigma, y, J){
  theta_hat <- (mu/tau^2 + y/sigma^2)/(1/tau^2 + 1/sigma^2)
  V_theta <- 1/(1/tau^2 + 1/sigma^2)
  rnorm(J, theta_hat, sqrt(V_theta))
}
mu_update <- function(theta, tau, J){
  rnorm(1, mean(theta), tau/sqrt(J))
}
tau_update <- function(theta, mu, J){
  sqrt(sum((theta-mu)^2)/rchisq(1,J-1))
}
```

Eight schools: Gibbs sampler

Set up 5 chains with a smaller number of iterations (1000) first.

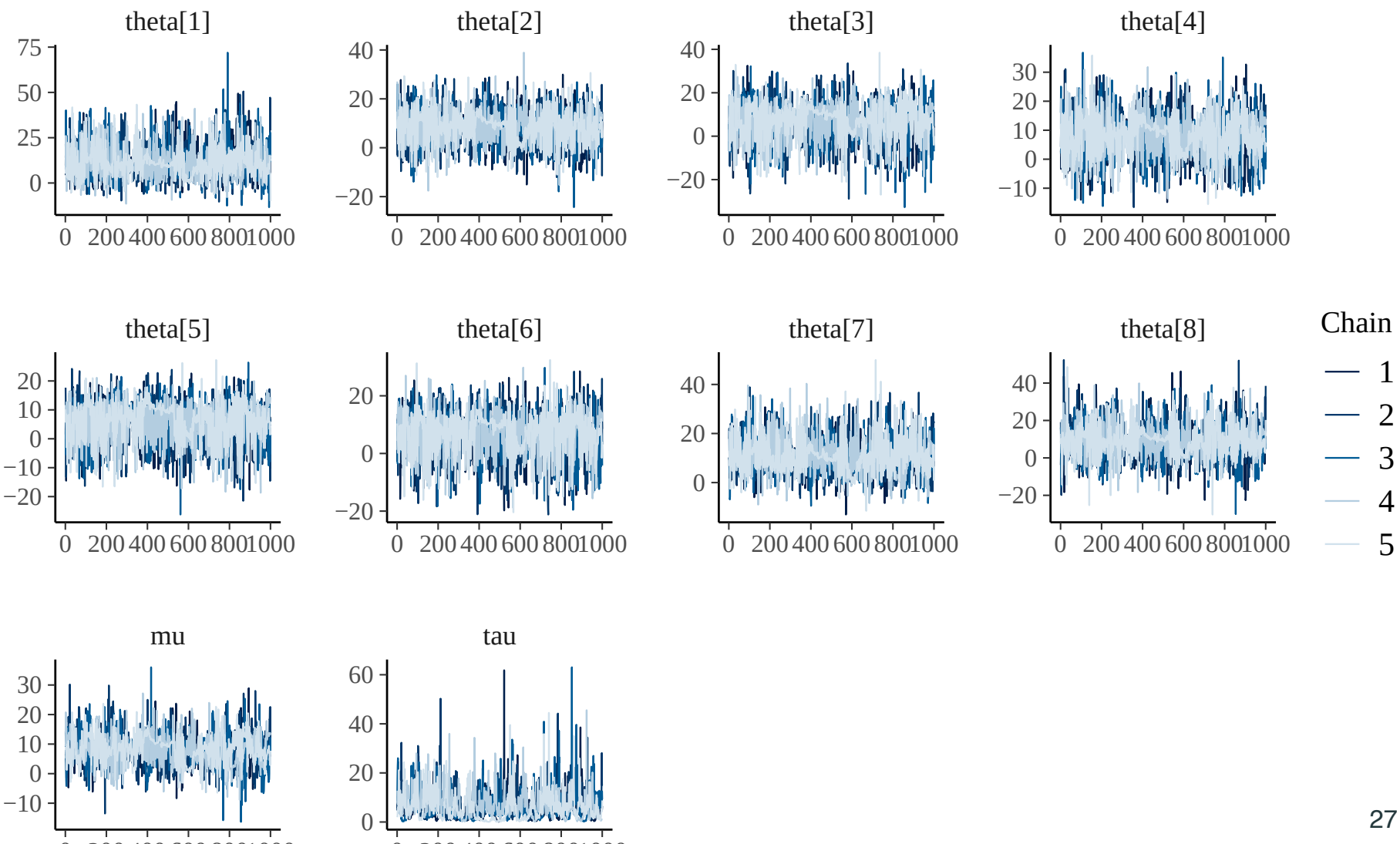
```
y <- c(28,8,-3,7,-1,1,18,12); s <- c(15,10,16,11,9,11,10,18); J <- 8
chains <- 5
iter <- 1000
sims <- array(NA, c(iter, chains, J+2))
dimnames(sims) <- list(NULL, NULL,
    c(paste("theta[", 1:8, "]", sep=""), "mu", "tau"))
for (m in 1:chains){
  mu <- rnorm(1, mean(y), sd(y))
  tau <- runif(1, 0, sd(y))
  for (t in 1:iter){
    theta <- theta_update(mu, tau, s, y, J)
    mu <- mu_update(theta, tau, J)
    tau <- tau_update(theta, mu, J)
    sims[t,m,] <- c(theta, mu, tau)
  }
}
library(rstan)
monitor(sims)
```

Eight schools: Gibbs sampler

```
> monitor(sims)
Inference for the input samples (5 chains: each with iter = 1000; warmup = 500):

          Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
theta[1]  0.1 10.5 27.1 11.5 8.5  1.01      298      867
theta[2] -2.2  7.6 17.6  7.6 6.2  1.01      456     1191
theta[3] -8.2  6.1 17.3  5.8 7.7  1.01      538     1268
theta[4] -3.0  7.5 17.6  7.5 6.3  1.01      426     1375
theta[5] -6.5  5.2 14.1  4.7 6.4  1.01      461     1031
theta[6] -6.5  6.4 16.1  5.8 6.9  1.01      466     1438
theta[7]  0.4 10.1 22.2 10.5 6.7  1.01      308      911
theta[8] -3.2  8.1 22.0  8.3 7.9  1.01      495     1144
mu       -0.5  7.5 15.4  7.6 5.1  1.02      256      560
tau       0.8  5.3 16.9  6.7 5.7  1.02      143      213

For each parameter, Bulk_ESS and Tail_ESS are crude measures of
effective sample size for bulk and tail quantities respectively (an ESS > 100
per chain is considered good), and Rhat is the potential scale reduction
factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

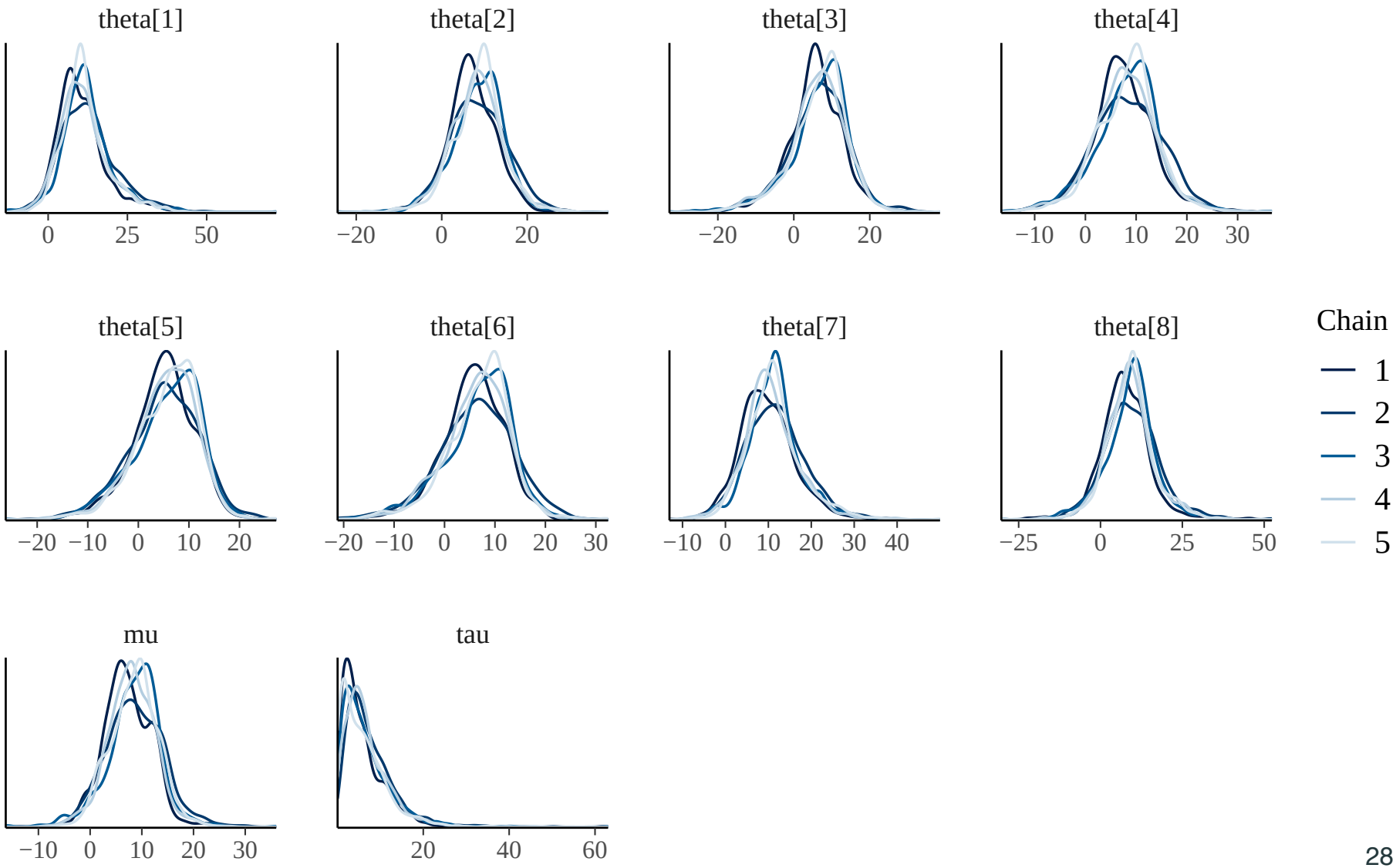
Eight schools: Gibbs sampler

```
library(bayesplot)
mcmc_trace(sims)
```



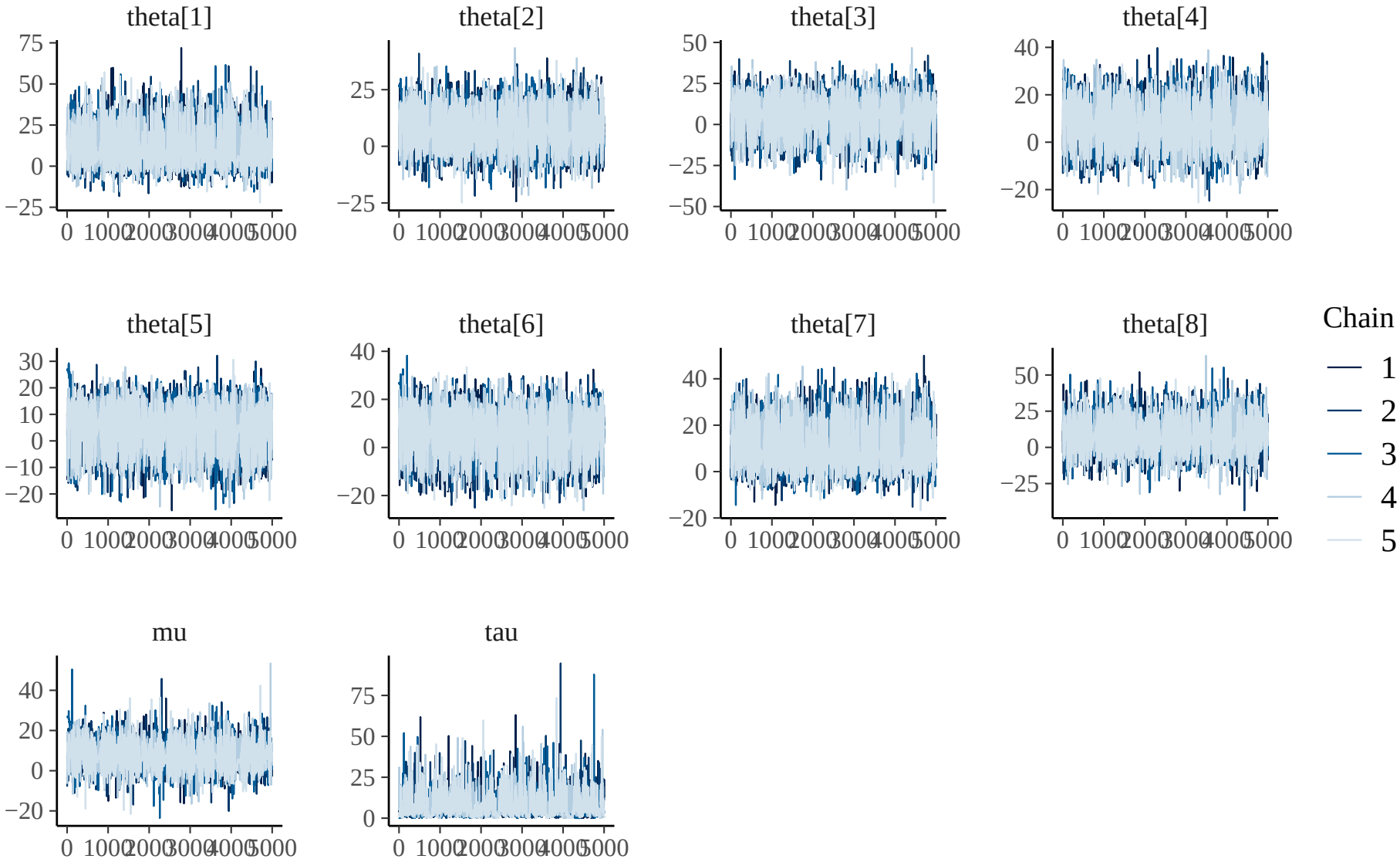
Eight schools: Gibbs sampler

```
mcmc_dens_overlay(sims)
```



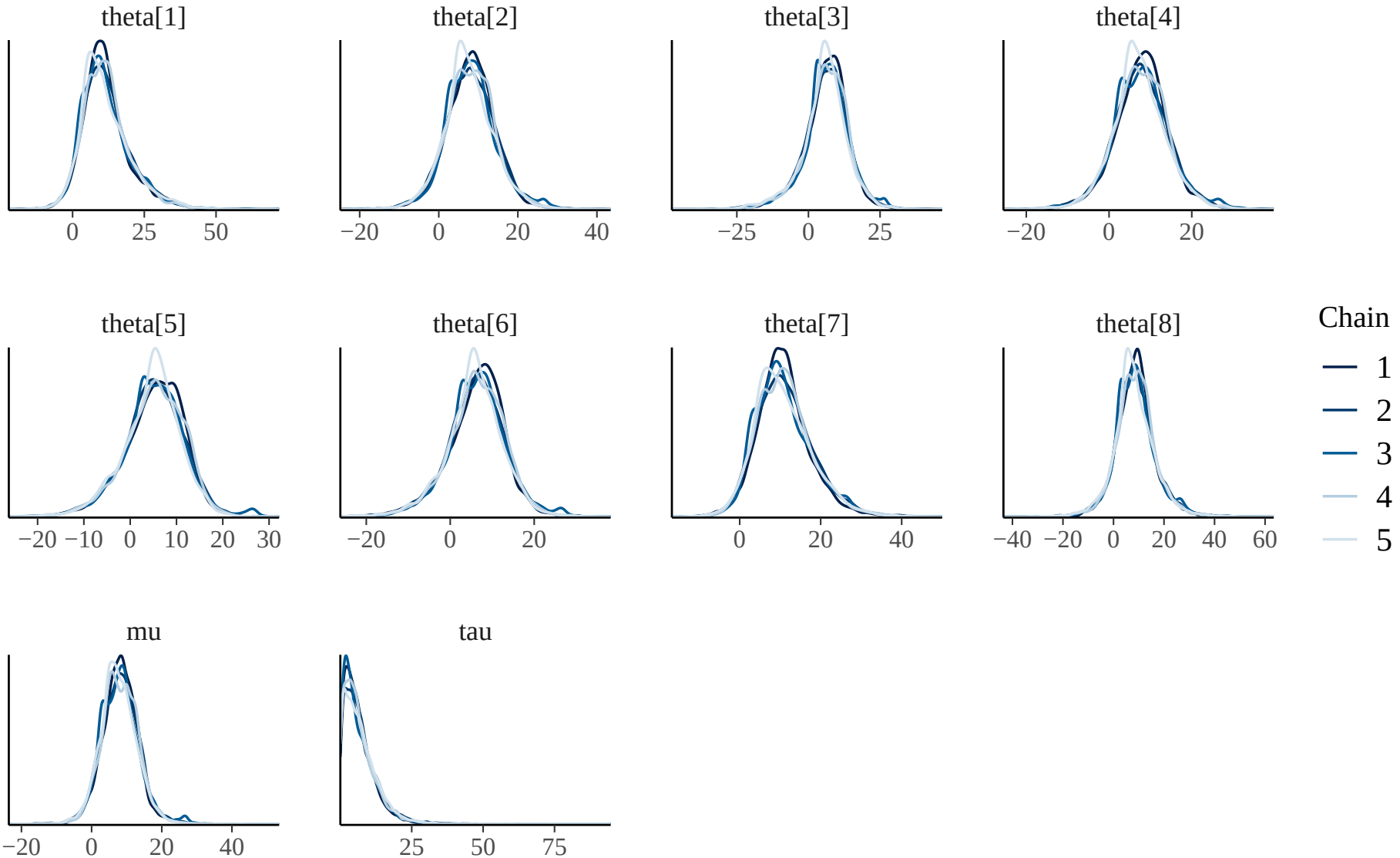
Eight schools: Gibbs sampler

The trace plot and the density plots from multiple chains does not seem too bad. It gets better if we increase the number of iterations to 5000.



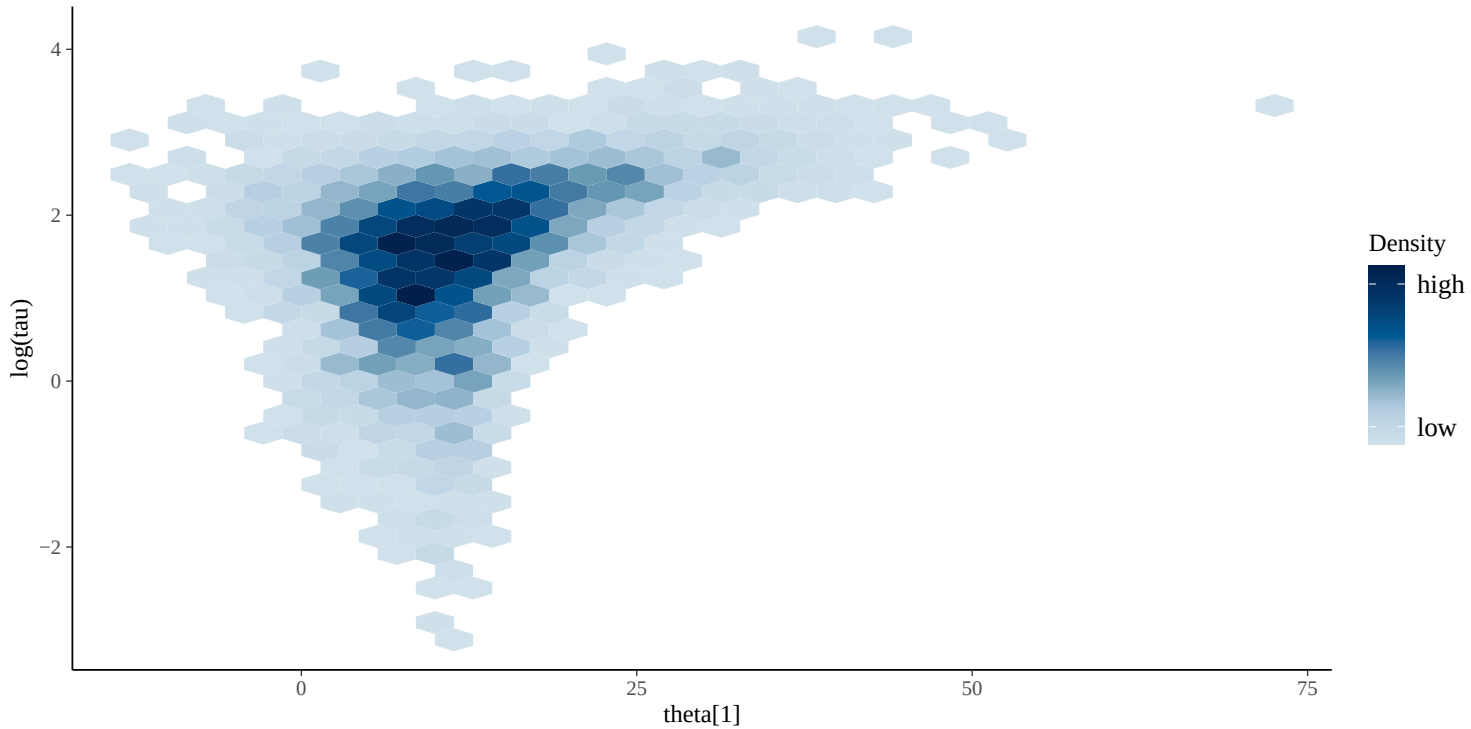
Eight schools: Gibbs sampler

The trace plot and the density plots from multiple chains does not seem too bad. It gets better if we increase the number of iterations to 5000.



Eight schools: Gibbs sampler

To see the potential problem we discussed earlier, let us look at the bivariate density plot for θ_1 and $\log(\tau)$.



Note: this type of "funnel" shape is very common in hierarchical models!

Eight schools: a reparameterized Gibbs sampler

Consider the following reparameterized model

$$\begin{aligned}\bar{y}_{\cdot j}|\theta_j &\sim N(\theta_j, \sigma_j^2)\\ \theta_j &= \mu + \alpha\gamma_j\\ \gamma_j &\sim N(0, \tilde{\tau})\end{aligned}$$

And prior $p(\mu, \alpha) \propto 1$. Notice that $\tilde{\tau}$ is over-parameterized and we can set it to 1. Then τ in the previous model is equivalent to $\tau = |\alpha|$. We can proceed with the following Gibbs sampler for this non-centered representation:

- Sample $\gamma_j|y, \mu, \alpha, \tilde{\tau}, \sigma \sim N(\frac{\alpha(y_j-\mu)/\sigma_j^2}{1/\tilde{\tau}^2+\alpha^2/\sigma_j^2}, (1/\tilde{\tau}^2 + \alpha^2/\sigma_j^2)^{-1})$.
- Sample $\alpha|y, \mu, \gamma, \tilde{\tau}, \sigma \sim N(\frac{\sum_j \gamma_j(y_j-\mu)/\sigma_j^2}{\sum_j \gamma_j^2/\sigma_j^2}, (\sum_j \gamma_j^2/\sigma_j^2)^{-1})$
- Sample $\mu|y, \alpha, \gamma, \sigma \sim N(\frac{\sum_j (y_j-\alpha\gamma_j)/\sigma_j^2}{\sum_j 1/\sigma_j^2}, (\sum_j 1/\sigma_j^2)^{-1})$.

Eight schools: Gibbs sampler (non-centered model)

```
gamma_update <- function(alpha, y, mu, sigma, tau){
  gamma_hat <- (alpha*(y-mu)/sigma^2)/(1/tau^2 + alpha^2/sigma^2)
  V_gamma <- 1/(1/tau^2 + alpha^2/sigma^2)
  rnorm(J, gamma_hat, sqrt(V_gamma))
}
alpha_update <- function(gamma, y, mu, sigma, tau){
  alpha_hat <- sum(gamma*(y-mu)/sigma^2)/sum(gamma^2/sigma^2)
  V_alpha <- 1/sum(gamma^2/sigma^2)
  rnorm(1, alpha_hat, sqrt(V_alpha))
}
mu_update2 <- function(y, alpha, gamma, sigma){
  mu_hat <- sum((y-alpha*gamma)/sigma^2)/sum(1/sigma^2)
  V_mu <- 1/sum(1/sigma^2)
  rnorm(1, mu_hat, sqrt(V_mu))
}
```

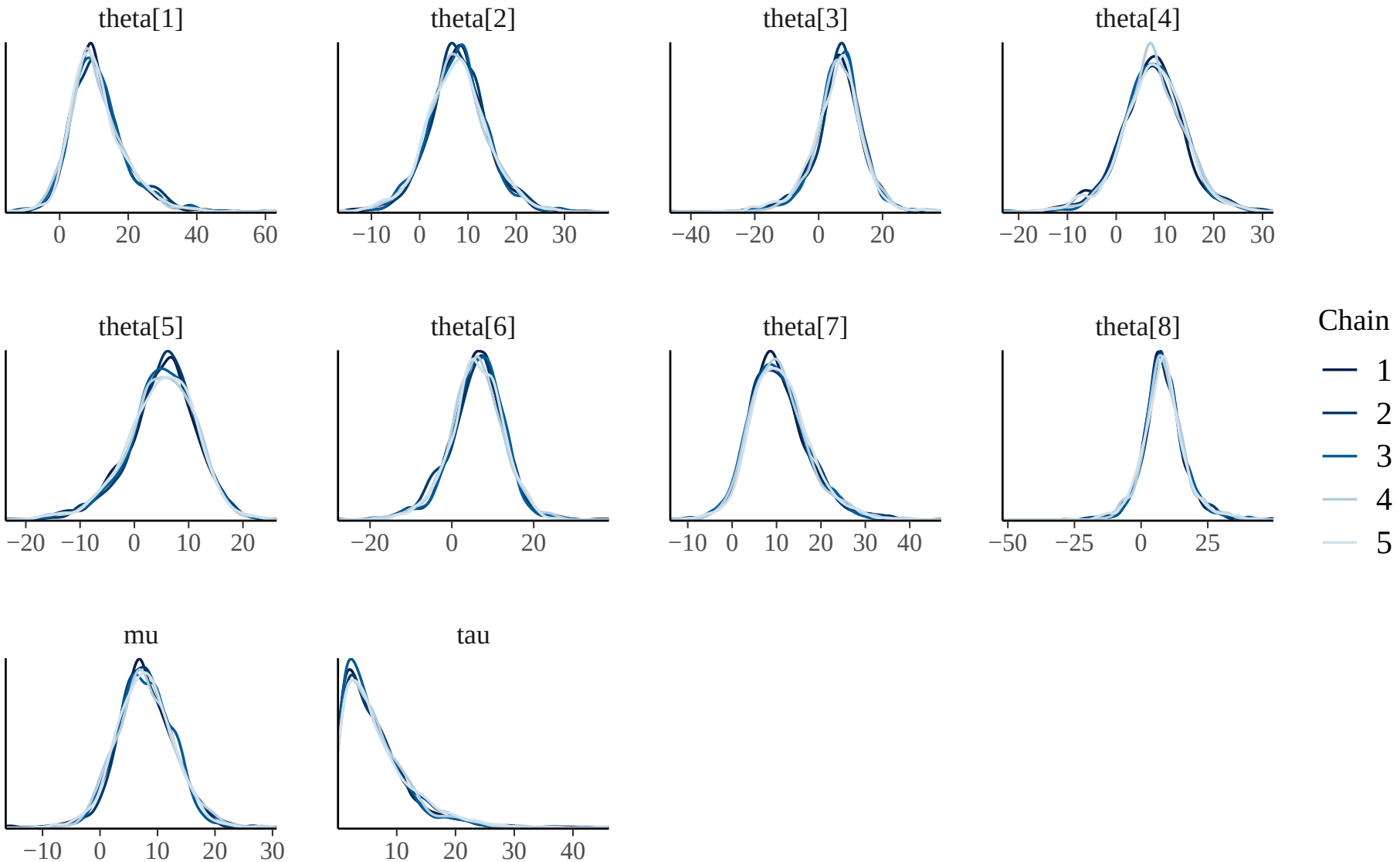
Eight schools: Gibbs sampler (non-centered model)

```
sims2 <- array(NA, c(iter, chains, J+2))
dimnames(sims2) <- list(NULL, NULL,
c(paste("theta[", 1:8, "]", sep=""), "mu", "tau"))
for (m in 1:chains){
  alpha <- 1
  mu <- rnorm(1, mean(y), sd(y))
  tau <- 1
  for (t in 1:iter){
    gamma <- gamma_update(alpha, y, mu, s, tau)
    alpha <- alpha_update(gamma, y, mu, s, tau)
    mu <- mu_update2(y, alpha, gamma, s)
    sims2[t,m,] <- c(mu + alpha*gamma, mu, abs(alpha)*tau)
  }
}
monitor(sims2)
```

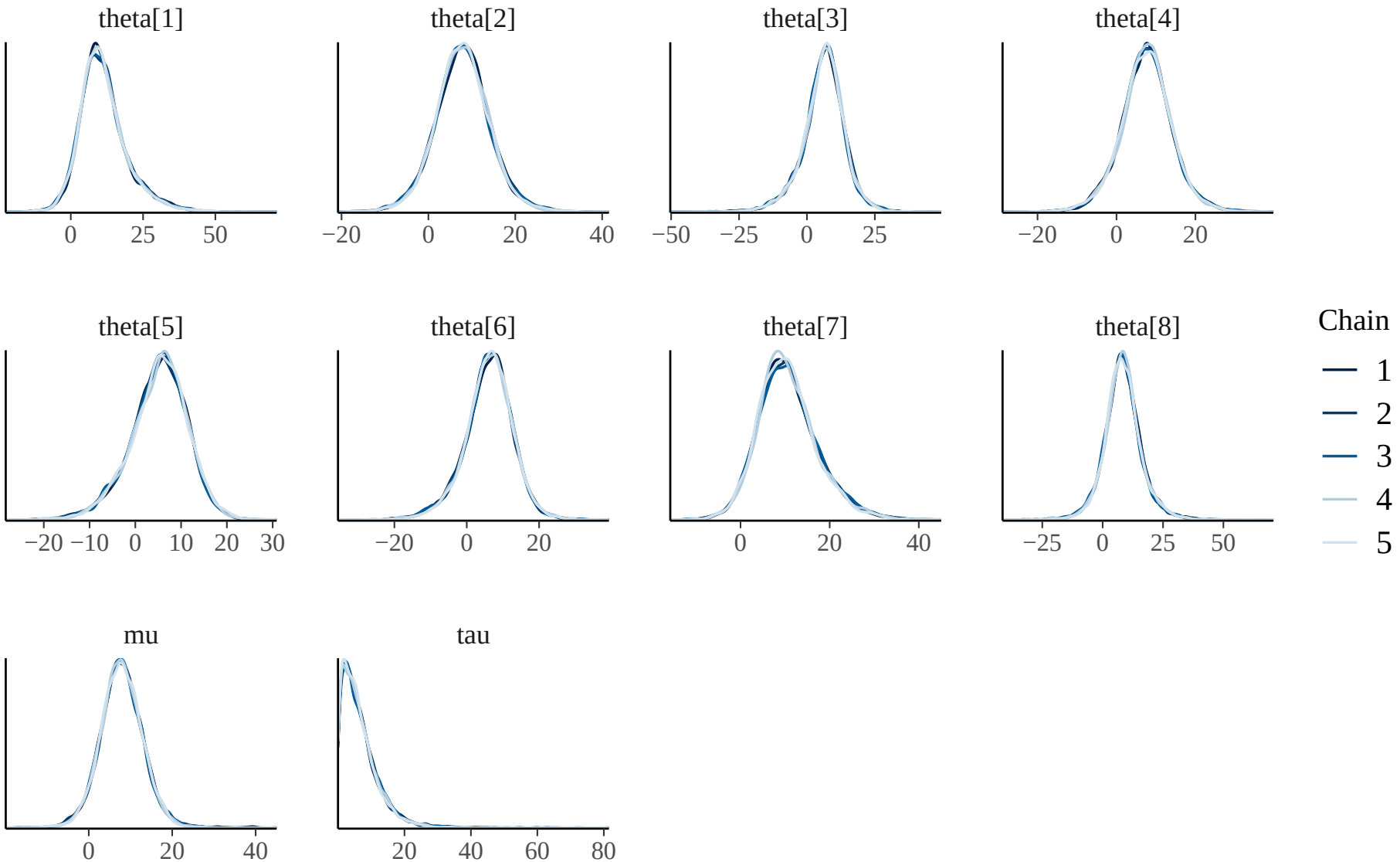
Eight schools: Gibbs sampler (non-centered model)

```
Inference for the input samples (5 chains: each with iter = 1000; warmup = 500):

           Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
theta[1] -0.2  9.9 26.7 11.1 8.3     1     1817     1649
theta[2] -1.9  8.0 18.1  8.0 6.2     1     2692     2530
theta[3] -7.2  6.4 17.7  5.9 7.9     1     1829     1849
theta[4] -2.9  7.7 18.2  7.7 6.5     1     2604     2442
theta[5] -6.1  5.6 14.5  5.1 6.3     1     2148     2211
theta[6] -5.0  6.6 16.4  6.3 6.7     1     2579     2327
theta[7]  0.8 10.0 23.0 10.6 6.8     1     2260     2437
theta[8] -3.5  8.1 22.8  8.7 8.1     1     1876     1886
mu       -0.1  7.7 16.0  7.9 5.0     1     1200     1450
tau       0.5  5.0 17.9  6.5 5.7     1     1230     1272

For each parameter, Bulk_ESS and Tail_ESS are crude measures of
effective sample size for bulk and tail quantities respectively (an ESS > 100
per chain is considered good), and Rhat is the potential scale reduction
factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

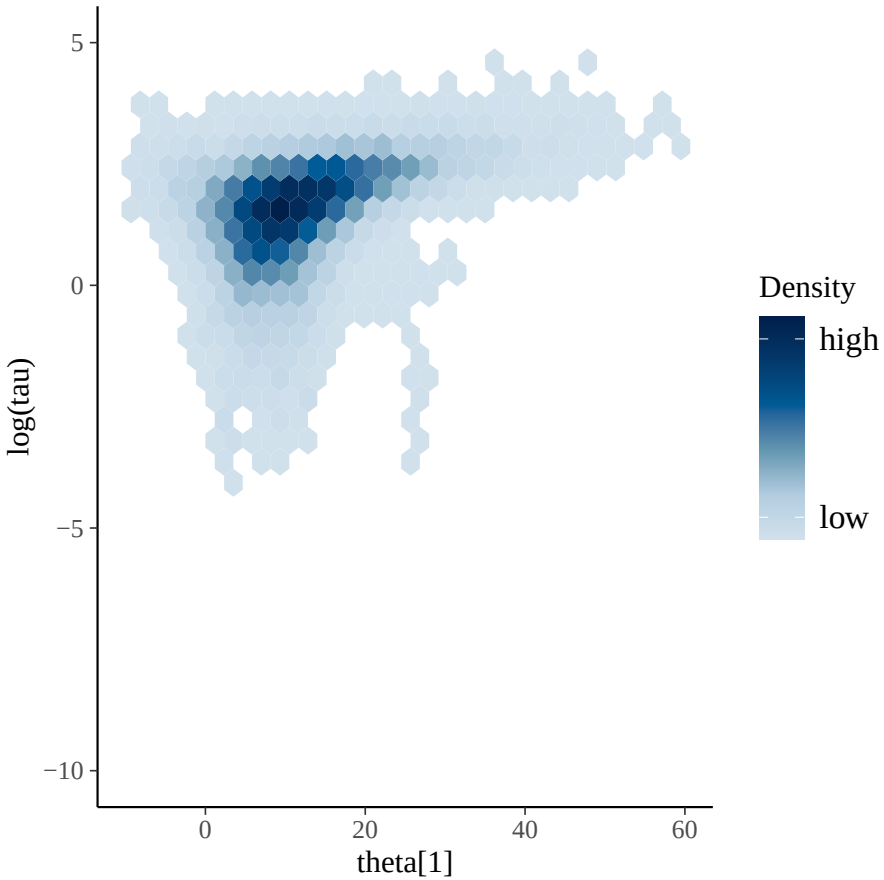
Eight schools: Gibbs sampler (non-centered model, 1000 iterations)

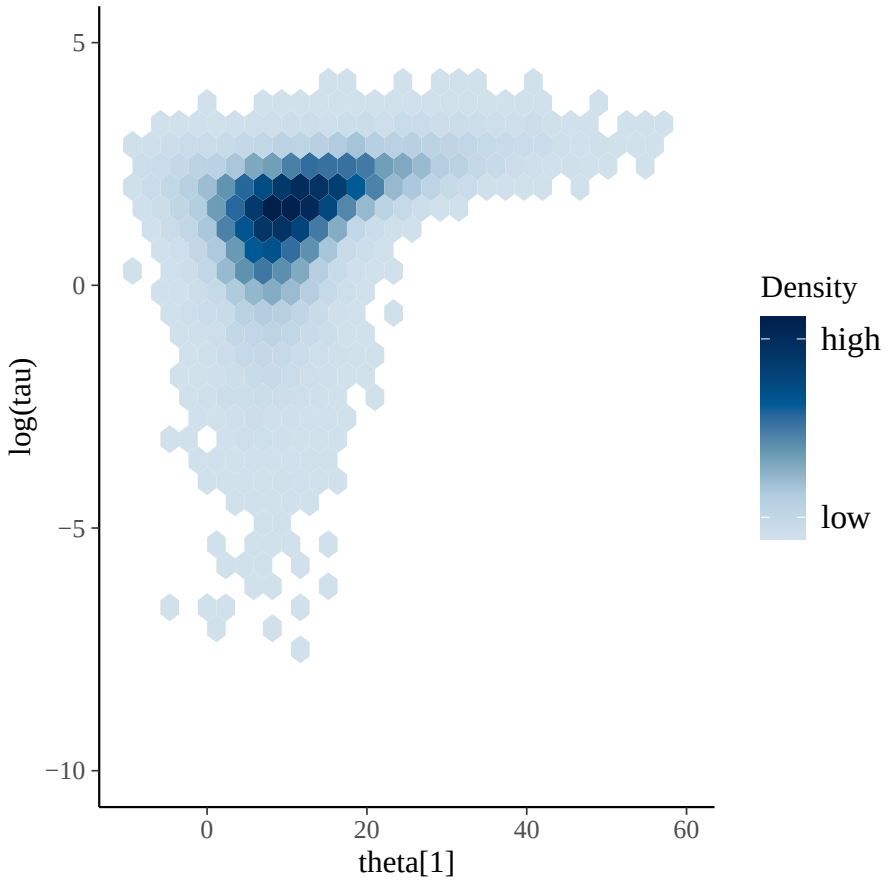


Eight schools: Gibbs sampler (non-centered model, 5000 iterations)

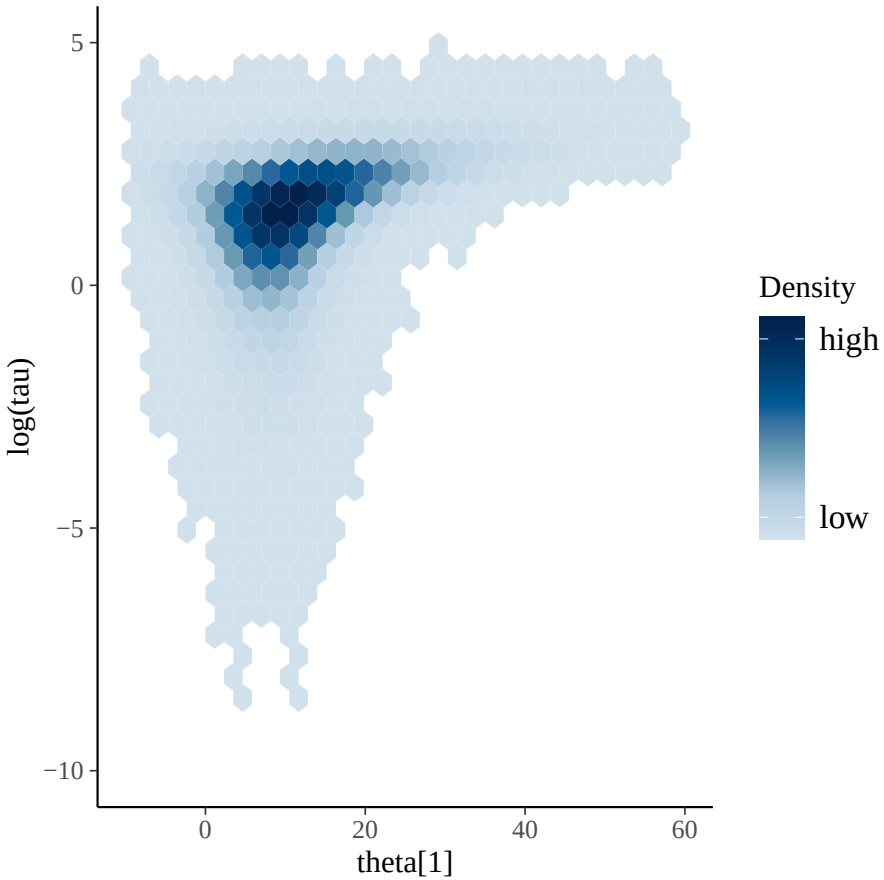


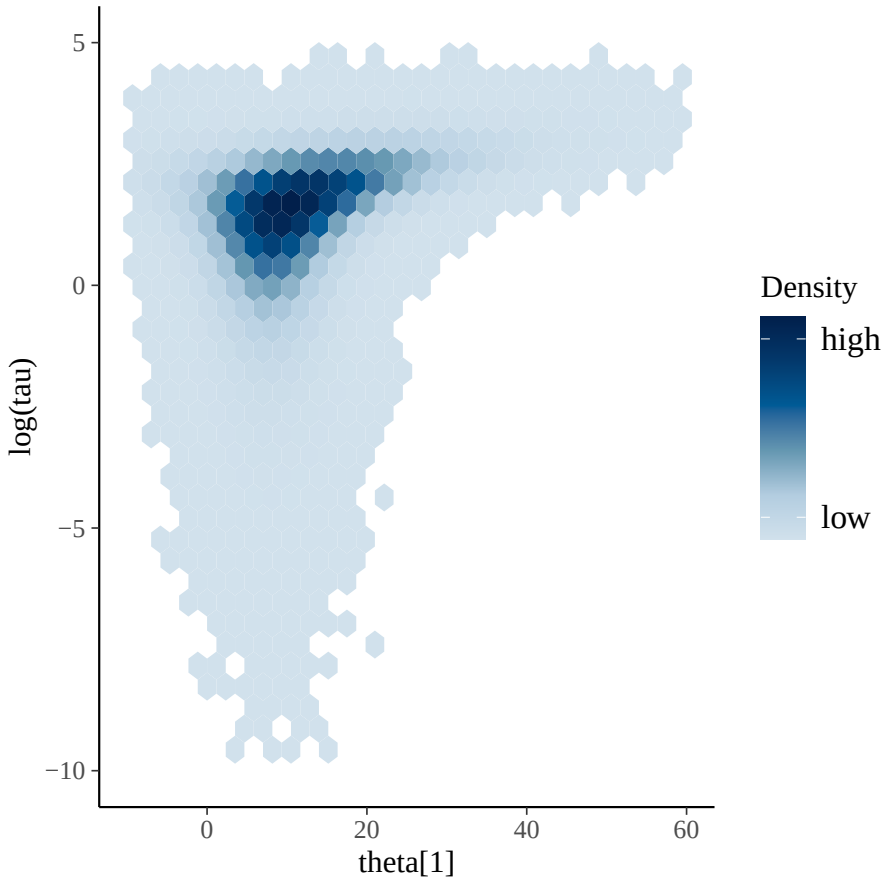
Eight schools: comparing the posterior (5,000 iterations)





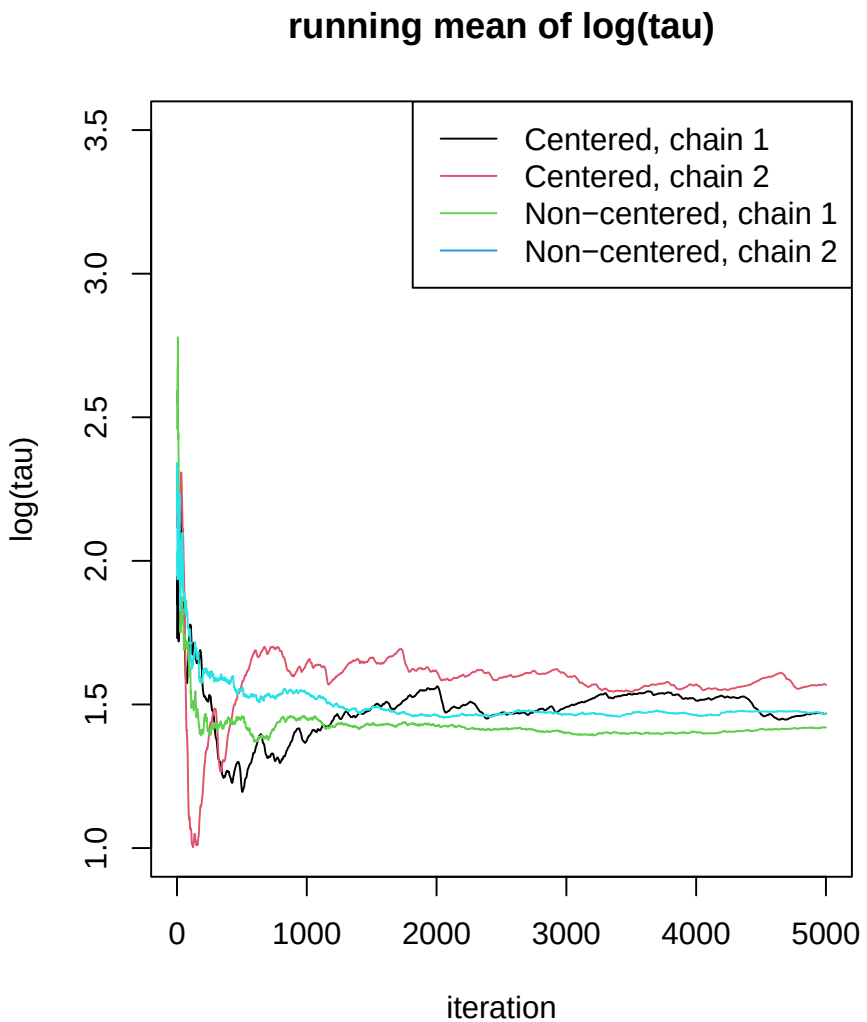
Eight schools: comparing the posterior (200,000 iterations)





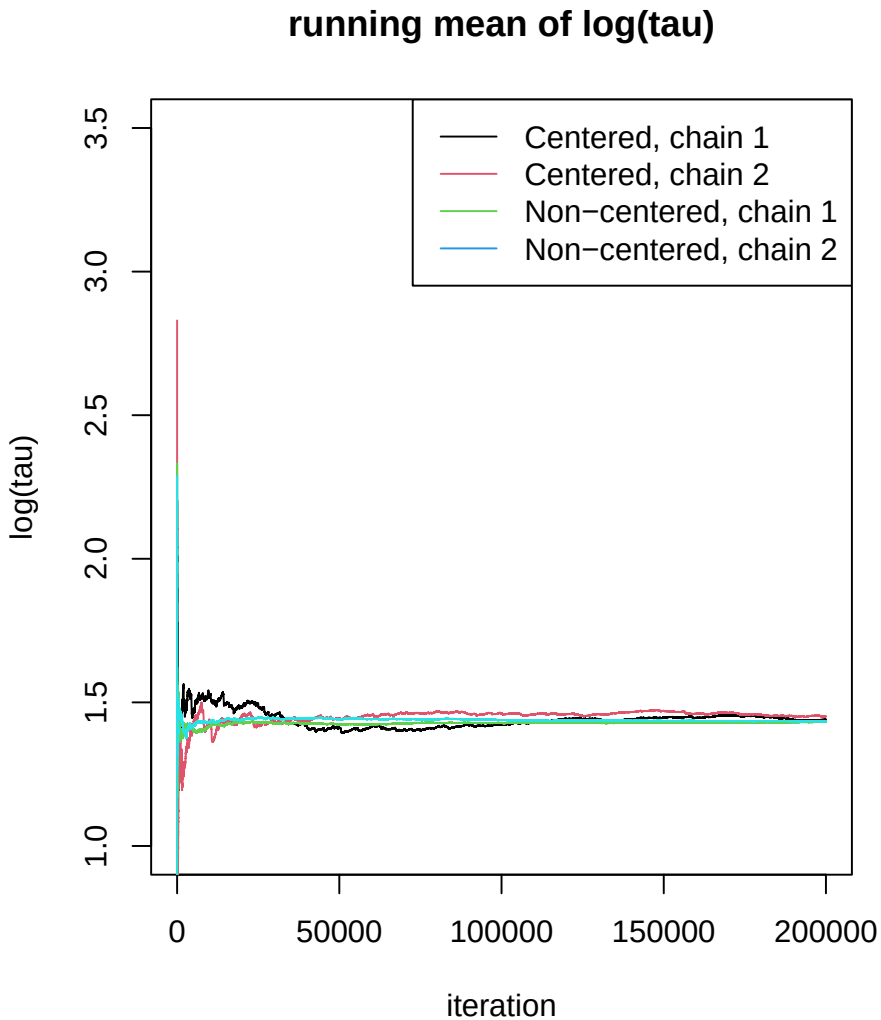
Eight schools: comparing the posterior (5,000 iterations)

We compute the running mean of $\log(\tau)$ over the iterations, we see a bias for the naive parameterizations after 5000 iterations, as the chains do not explore the small τ region very well.



Eight schools: comparing the posterior (200,000 iterations)

The running mean is ok with very long chains.

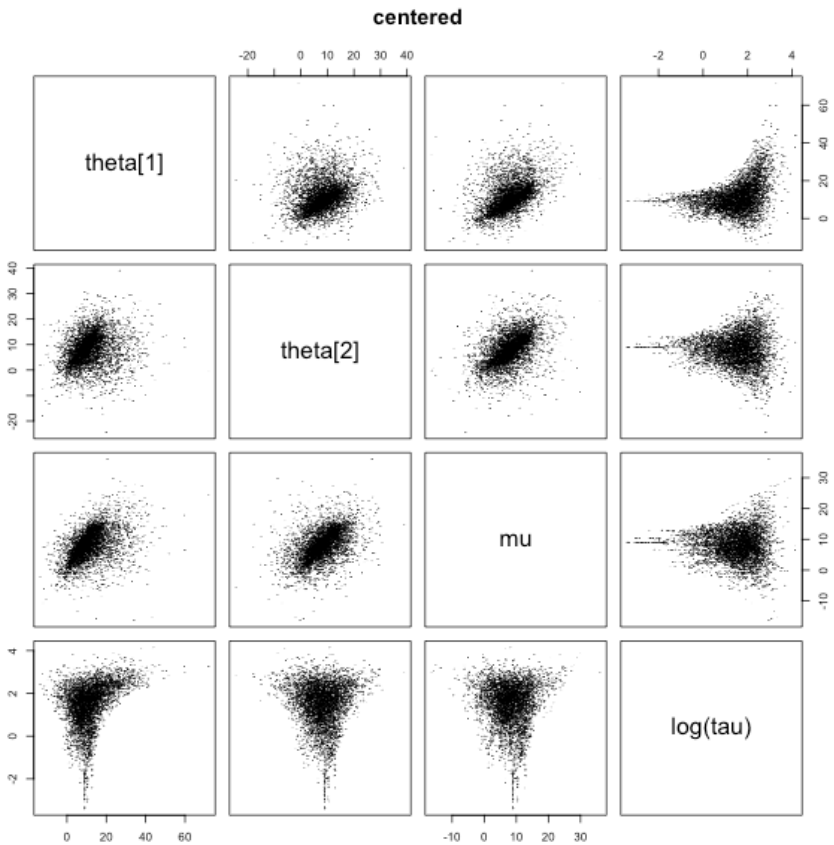


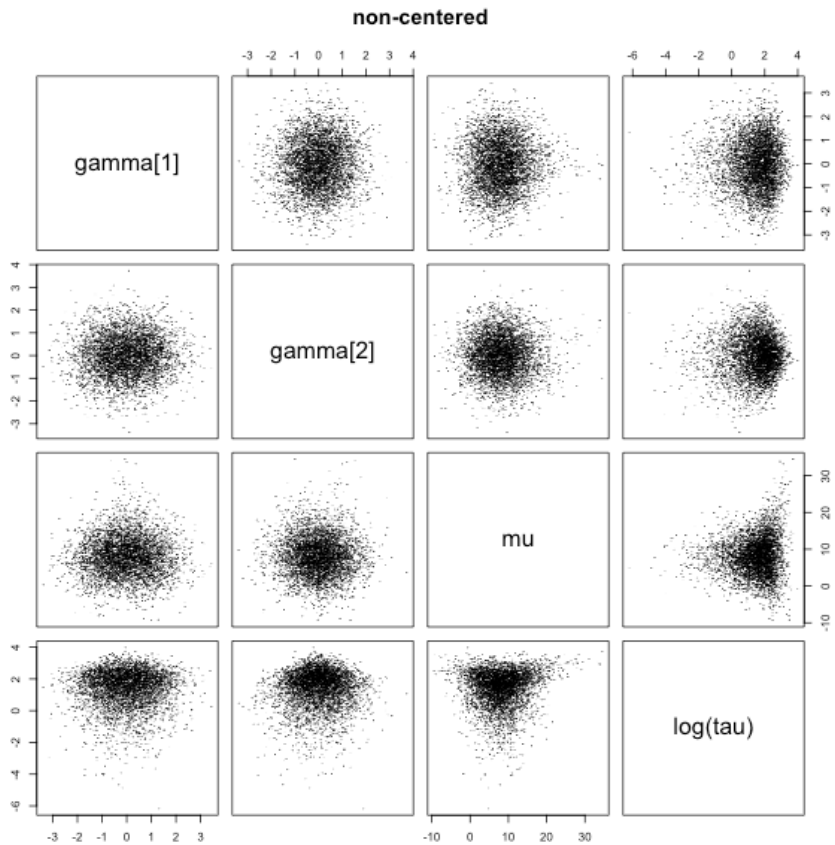
Centered v.s. non-centered parameterizations

- We expect the funnel shape of the posterior on the space of (τ, θ) as when τ will be very small when all of the θ's are close to μ, i.e., under strong shrinkage.
- The reparameterization reduces such correlation by considering the parameterization of (τ, γ).
- Why we care about the region where τ is small? It is because we know that the likelihood contribution is quite flat (i.e., σ_j^2 are large), so we expect stronger shrinkage.
- However, if the likelihood contribution is strong, we would expect θ_j to be closer to $\bar{y}_{\cdot j}$ instead of μ. In this case, the non-centered parameterization becomes more problematic as the deviation γ_j becomes more correlated with μ!
- Thus the choice of parameterization or implementation of the sampler should be understood through the problem and data likelihood.

Eight schools: posterior scatter plot, $\sigma^2 = \sigma^2_{original}$

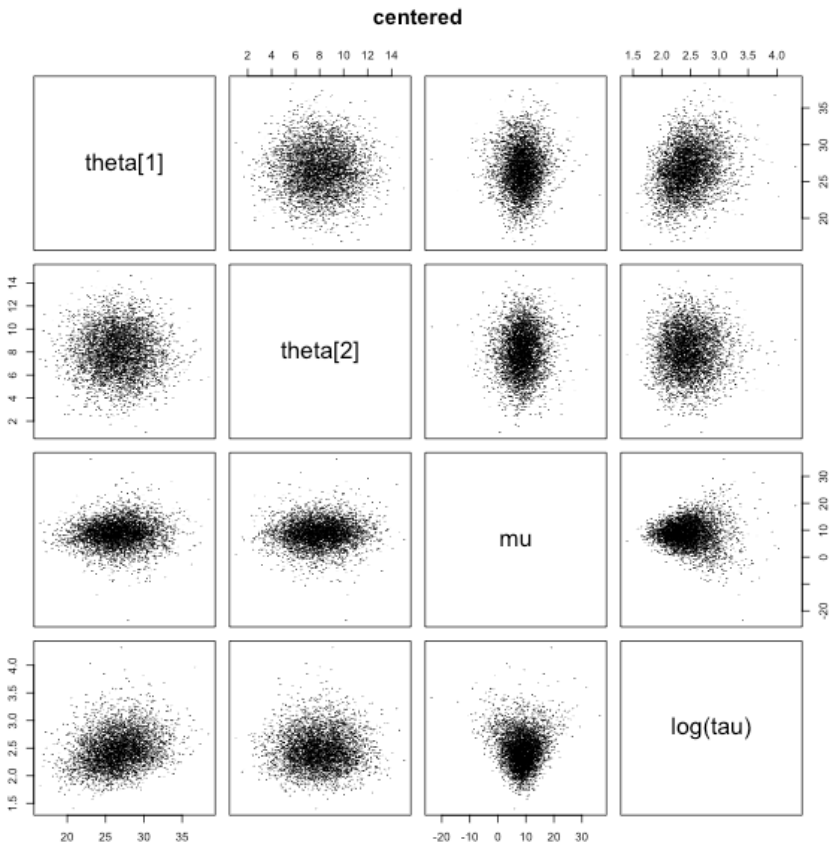
Funnel shape posterior in the original problem.

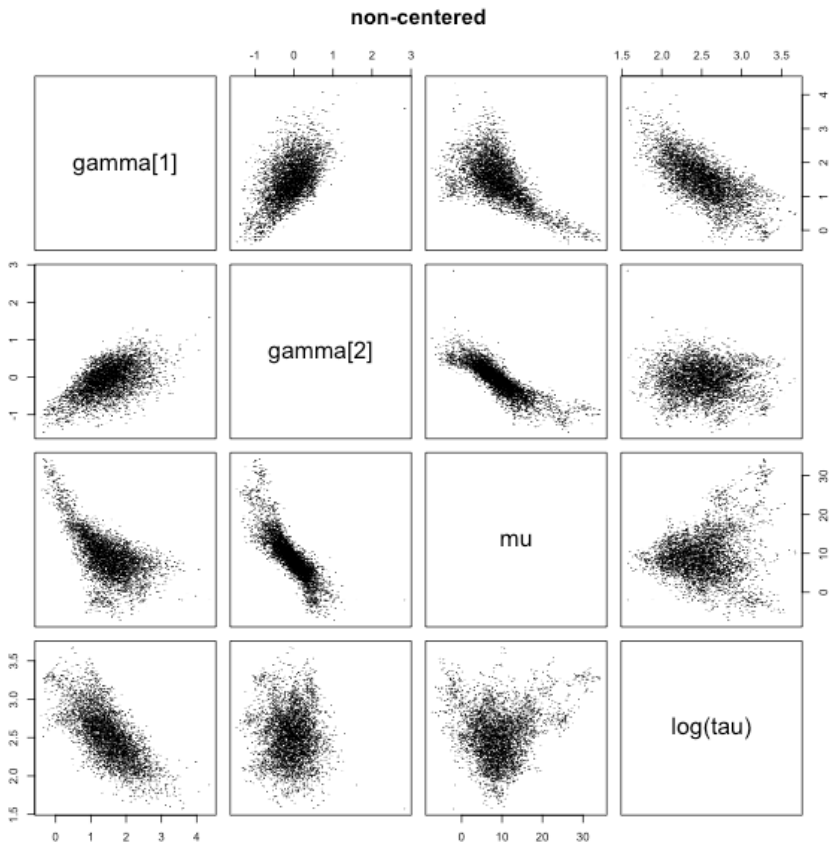




Eight schools: posterior scatter plot, $\sigma^2 = \sigma^2_{original}/5$

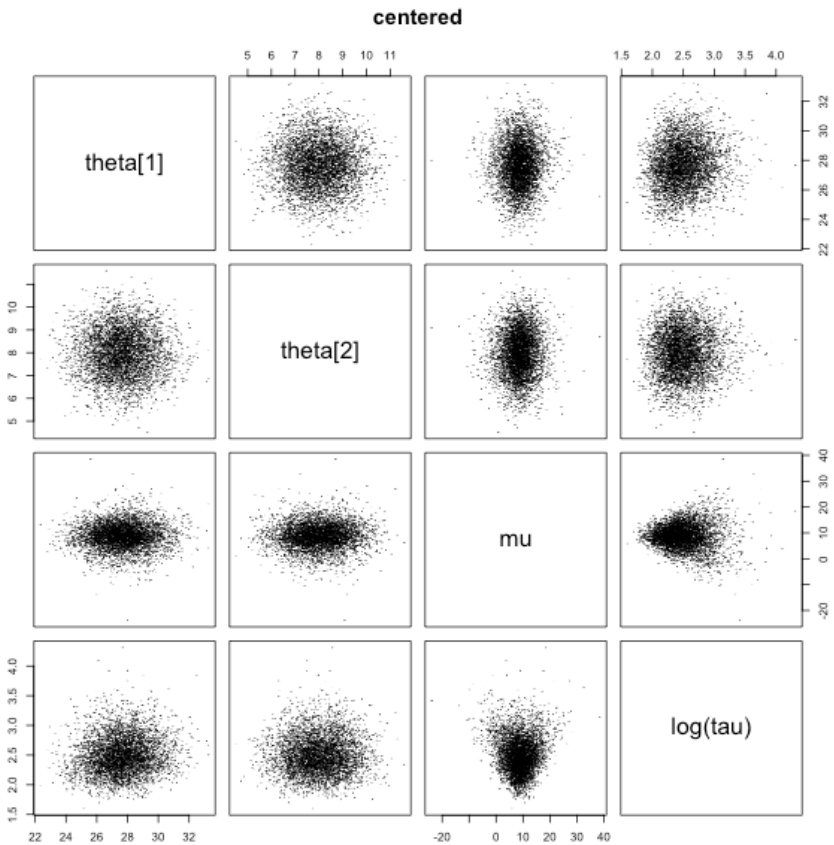
When likelihood becomes stronger.

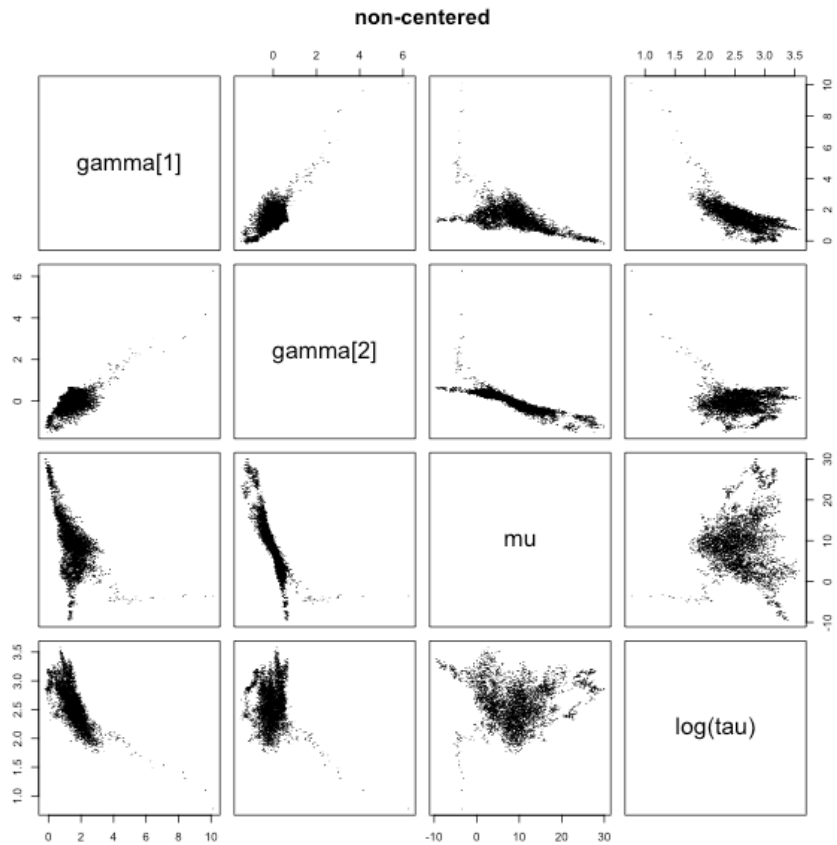




Eight schools: posterior scatter plot, $\sigma^2 = \sigma^2_{original}/10$

When likelihood becomes even stronger.





Partially collapsed sampler

- Another direction to improve the performance of MCMC is via marginalization or *collapsing*.
- Basically, it involves integrating a joint posterior distribution over a subset of unknown quantities.
- Consider a simple example with $(X, Y, Z) \sim N(0, \Sigma)$. Which of the following three samplers correctly sample from the distribution?

Sampler 1:

1. Draw $X|Y, Z$.
2. Draw $Y|X, Z$.
3. Draw $Z|X, Y$.

Sampler 2:

1. Draw $X|Y, Z$.
2. Draw $Y|Z$.
3. Draw $Z|X, Y$.

Sampler 3:

1. Draw $Y|Z$.
2. Draw $X|Y, Z$.
3. Draw $Z|X, Y$.

Example: random effect model

- *Example from Van Dyk, David A., and Taeyoung Park. "Partially collapsed Gibbs samplers: Theory and methods." JASA, 2008*
- Consider a simple random effect model

$$y_{ij} = \theta_i + \epsilon_{ij}$$

  for $i = 1, ..., k$ and $j = 1, ..., n$.
- Let $\theta_i \sim N(\mu, \tau^2)$ and $\epsilon_{ij} \sim N(0, \sigma^2)$.
- The three samplers:

Sampler 1:

1. Draw $\theta|\mu, Y$.
2. Draw $\mu|\theta, Y$.

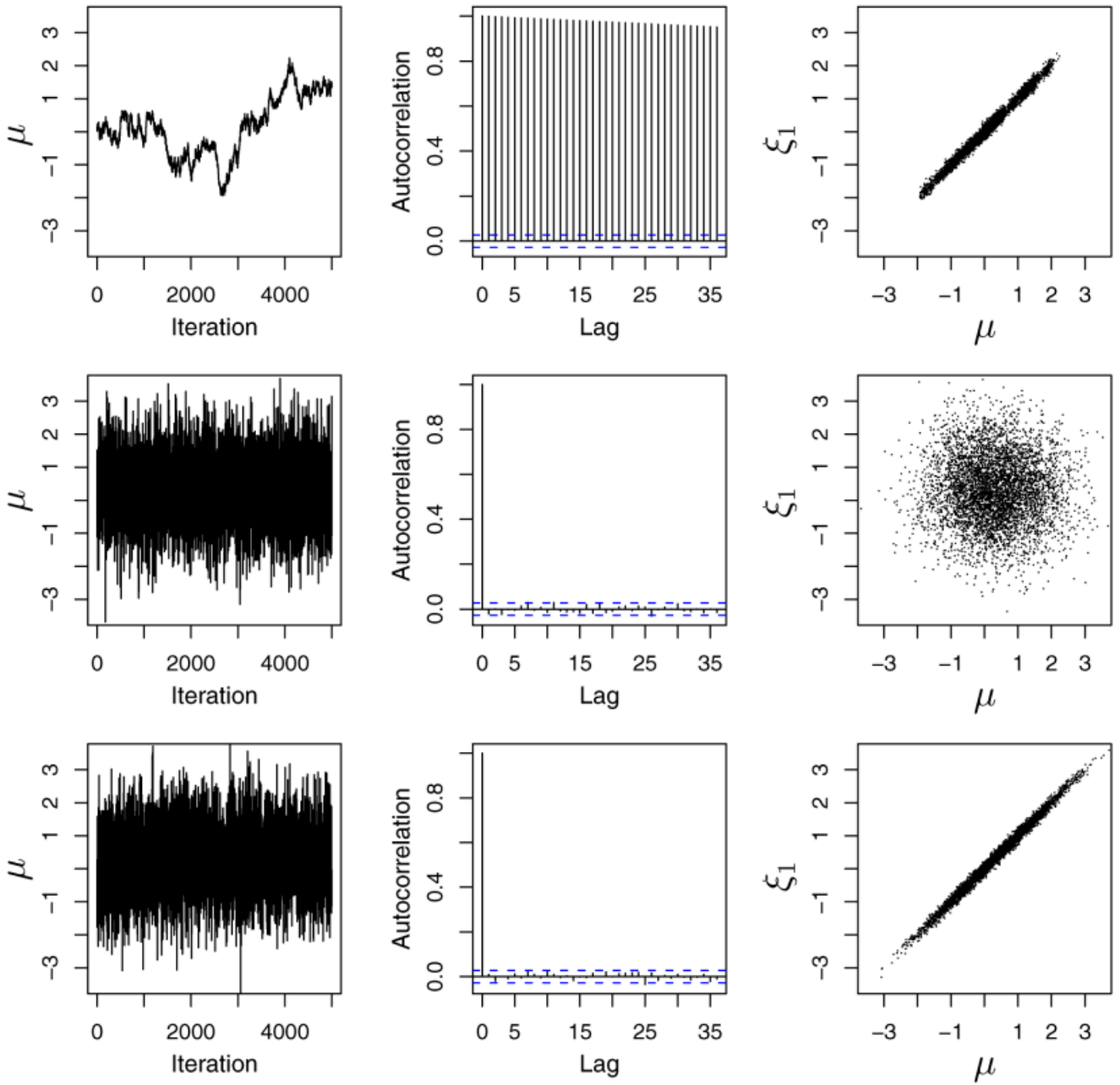
Sampler 2:

1. Draw $\theta|\mu, Y$.
2. Draw $\mu|Y$.

Sampler 3:

1. Draw $\mu|Y$.
2. Draw $\theta|\mu, Y$.

Example: random effect model



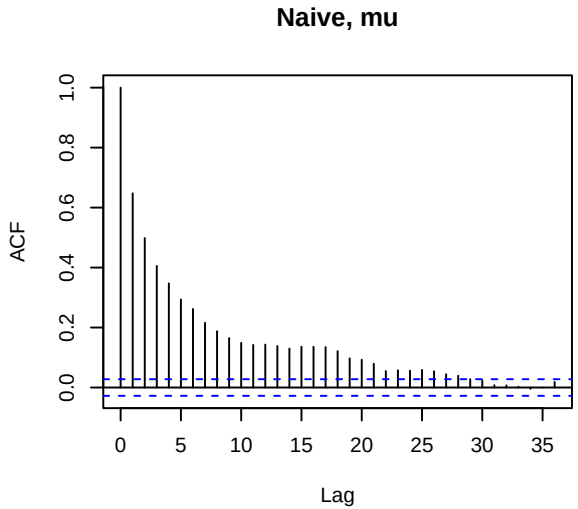
Eight schools: collapsed Gibbs sampler

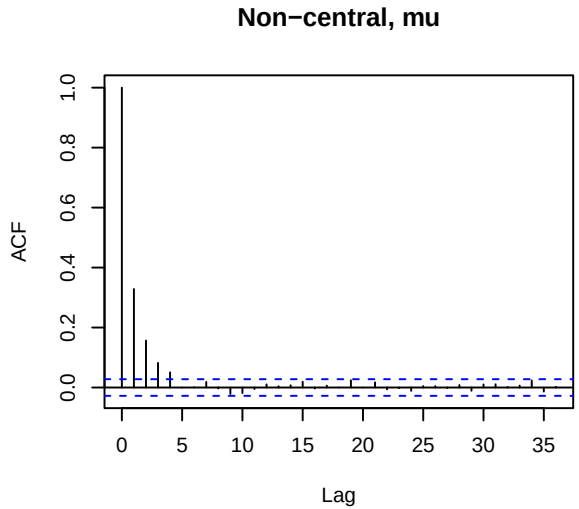
Return to the 8 school dataset, if we further collapse the sampling step for μ and draw $\mu|y, \tilde{\tau}, \sigma$ (the posterior was derived in the previous lecture).

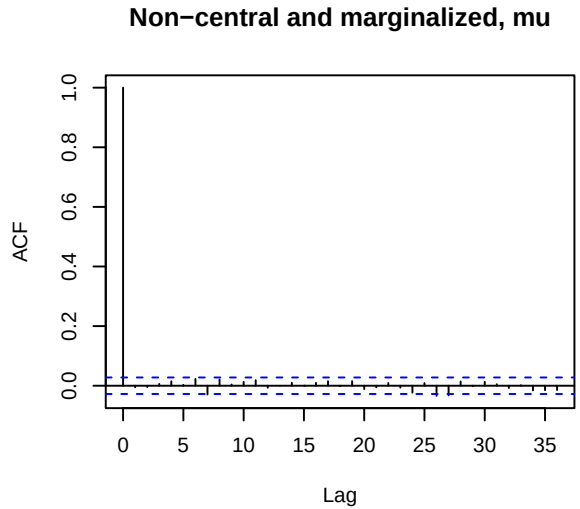
```
mu_update3 <- function(tau, sigma, y){
  mu0 <- sum(y / (tau^2 + sigma^2)) / sum(1 / (tau^2 + sigma^2))
  V0 <- 1/sum(1 / (tau^2 + sigma^2))
  rnorm(1, mu0, sqrt(V0))
}

sims3 <- array(NA, c(iter, chains, J+2))
dimnames(sims3) <- list(NULL, NULL,
    c(paste("theta[", 1:8, "]", sep=""), "mu", "tau"))
for (m in 1:chains){
  alpha <- 1
  mu <- rnorm(1, mean(y), sd(y))
  tau <- 1
  for (t in 1:iter){
    mu <- mu_update3(abs(alpha)*tau, s, y)
    gamma <- gamma_update(alpha, y, mu, s, tau)
    alpha <- alpha_update(gamma, y, mu, s, tau)
    sims3[t,m,] <- c(mu + alpha*gamma, mu, abs(alpha)*tau)
  }
}
```

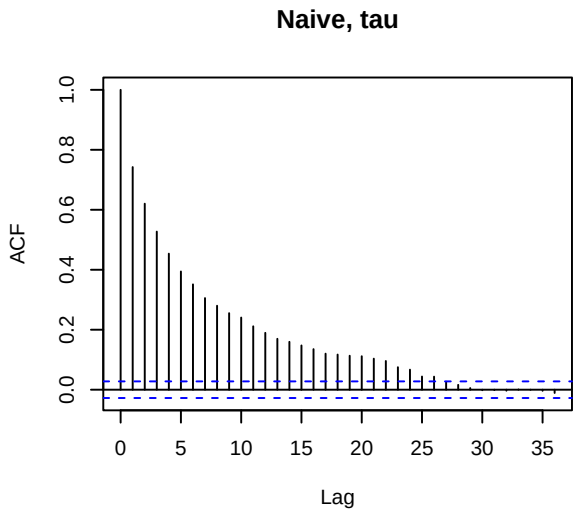
Eight schools: collapsed Gibbs sampler (5000 iterations)

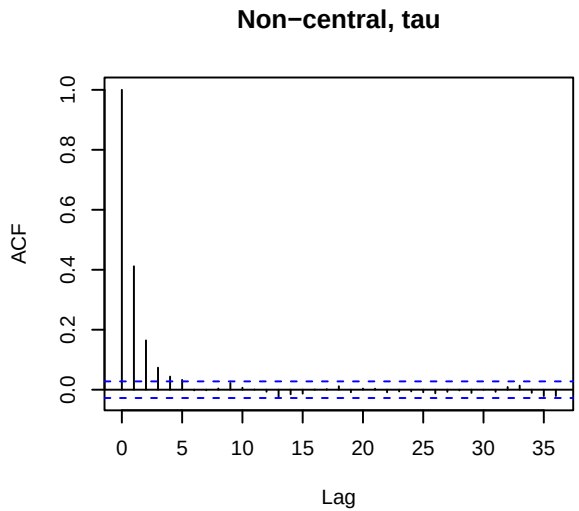
The autocorrelation is reduced by the marginalization step in sampling θ.

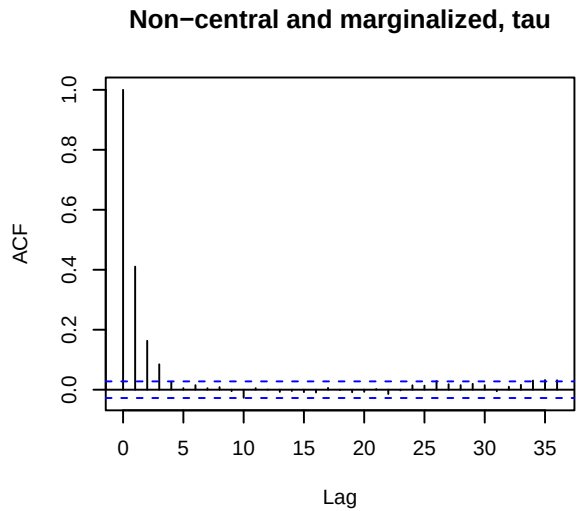












Convergence diagnostics

- In general, there is no definitive way to tell whether on ran a Markov chain long enough.
- Several useful diagnostic tools can illuminate problems with the sampler, bugs in the code, and suggest ways to improve the design of the MCMC sampler.
- *“There are known knowns. These are things we know that we know. There are known unknowns. That is to say, there are things that we now know we don’t know. But there are also unknown unknowns. These are things we do not know we don’t know.”*

Effective sample size

- In basic Monte Carlo with N samples, we know $\text{var}(\frac{1}{N}\sum_{i=1}^{N} X_i) = \text{var}(X_i)/N$.
- The samples $\{X_i\}$ from MCMC are not IID.

$$var(\frac{1}{N}\sum_{i=1}^{N} X_i) = \frac{1}{N^2}\sum_i \text{var}(X_i) + \frac{1}{N^2}\sum_i\sum_{j\neq i} Cov(X_i, X_j) = \frac{\text{var}(X_i)}{N_{eff}}$$

- Assume $\{X_i\}$ have the same distribution and are stationary, as $N \to \infty$,

$$N_{eff} \approx \frac{N}{1 + \sum_{t=1}^{\infty} \rho_t}$$

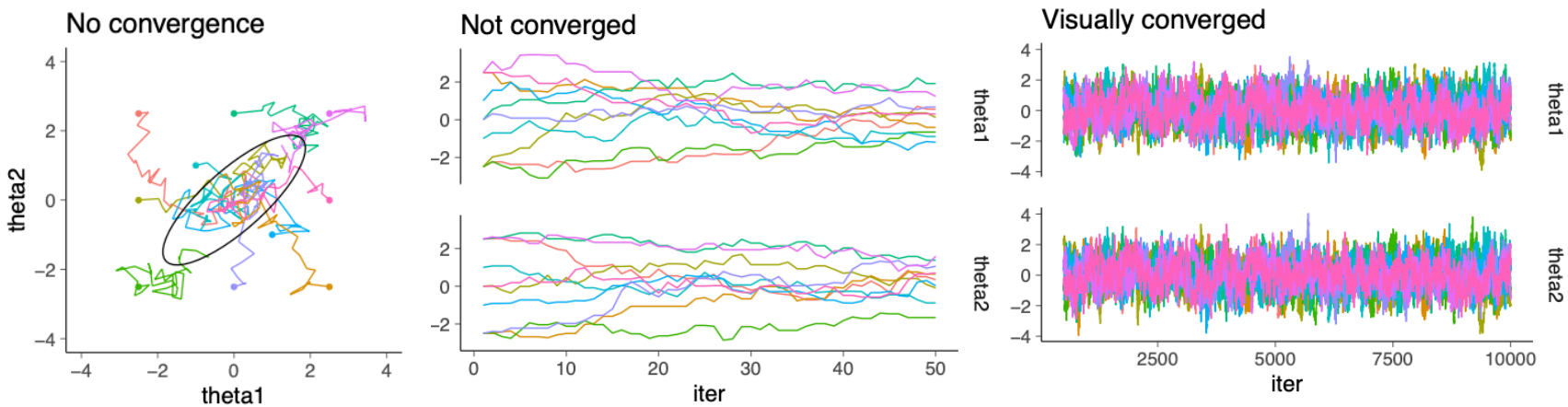
  where $\rho_t = Corr(X_i, X_{i+t})$.
- N_{eff} is usually called the 'effective sample size' (ESS).
- It measures the relative efficiency of the MCMC samples, i.e., how many IID samples from a basic Monte Carlo would achieve the same approximation error.
- ESS depends on the parameter of interest, e.g., the ESS for X_i^2 is usually different from X_i.
- There are several slightly different definitions of ESS. Note if you compute the ESS for multiple chains, instead of N, the numerator becomes MN.

Convergence diagnostics

- Visualizing MCMC output: Trace plots provide a useful method for detecting problems with MCMC convergence and mixing. Ideally, trace plots of unnormalized log posterior and model parameters should look like stationary time series. Slowly mixing Markov chains produce trace plots with high autocorrelation, which can be further visualized by autocorrelation plots at different lags. Slow mixing does not imply lack of convergence. Clearer to examine parameters transformed to $\mathbb{R}$.
- Comparing batches: We take two subsets of the MCMC output, say $[\theta^{(1)}, ..., \theta^{(T/2)}]$ and $[\theta^{(T/2+1)}, ..., \theta^{(T)}]$. If MCMC achieved stationarity at the time of collecting these batches, then both vectors follow the same stationary distribution.

Convergence diagnostics

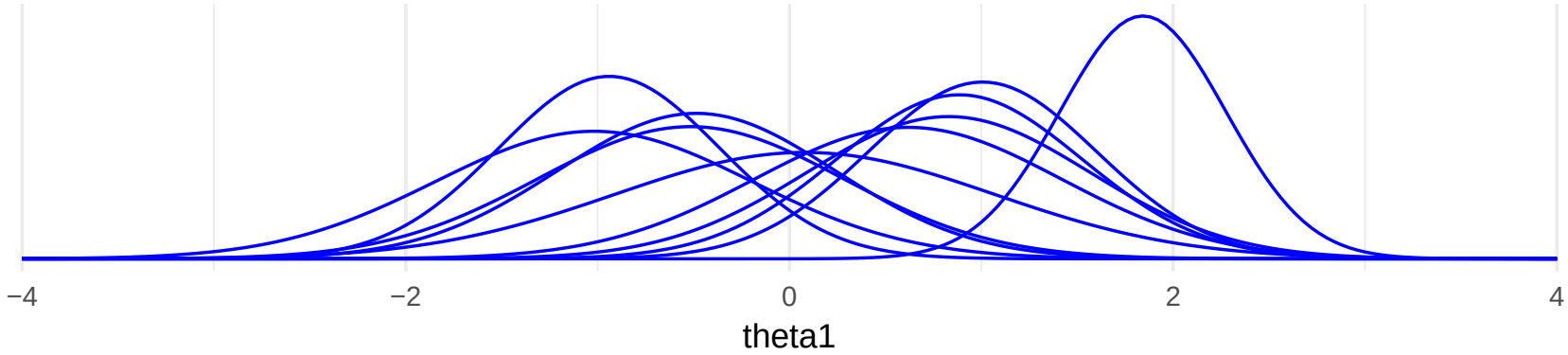
- Comparing multiple chains started from random initial conditions. There are many ways of performing such a comparison.



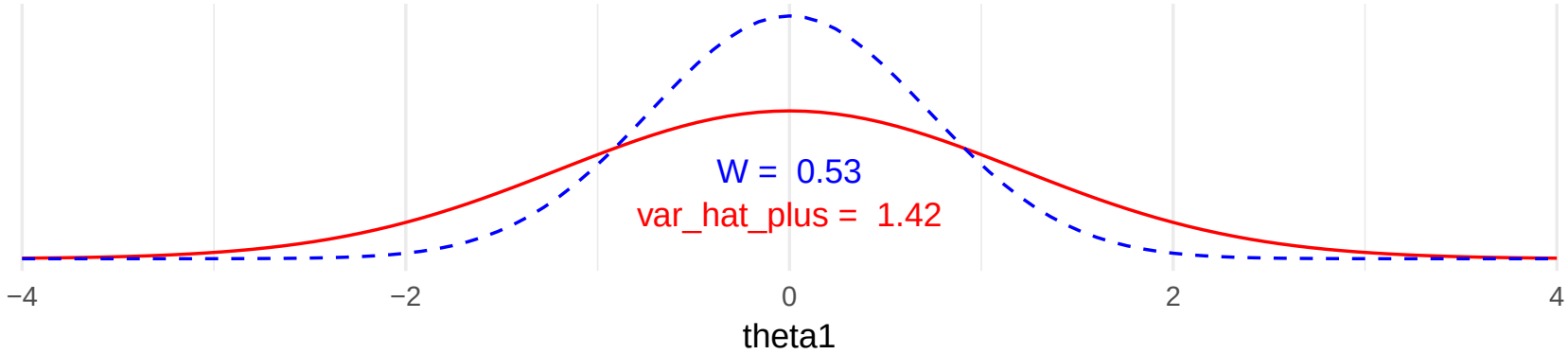
Gelman-Rubin diagnostics

- $\widehat{R}$ aka *potential scale reduction factor* (PSRF)
- The basic idea is we compare means and variances of the chains

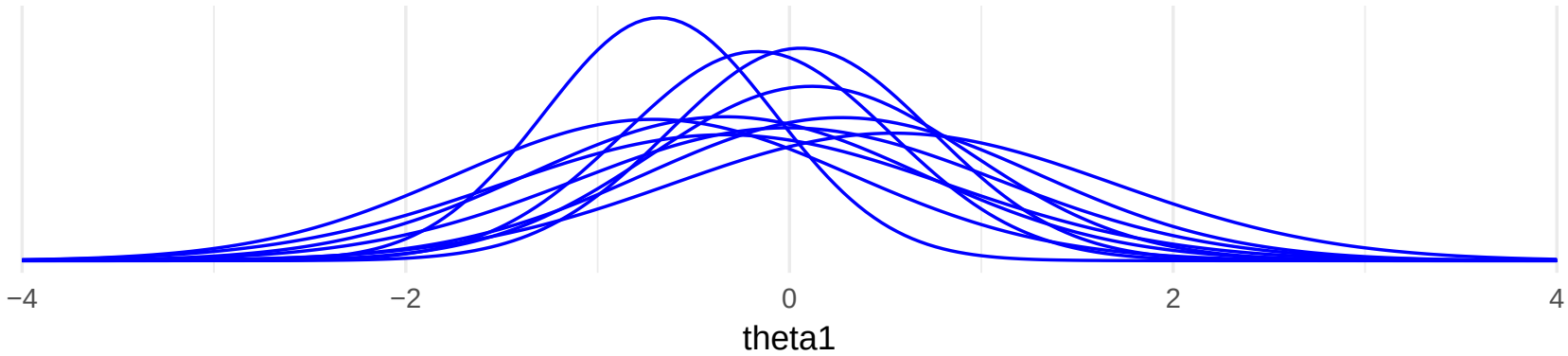
50 warmup, 50 post warmup iterations



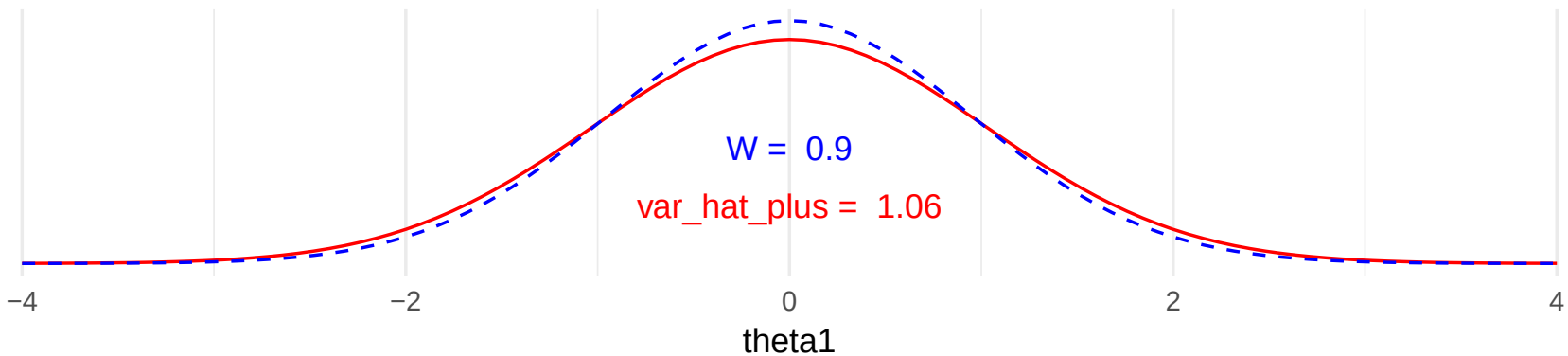
Rhat = 1.64



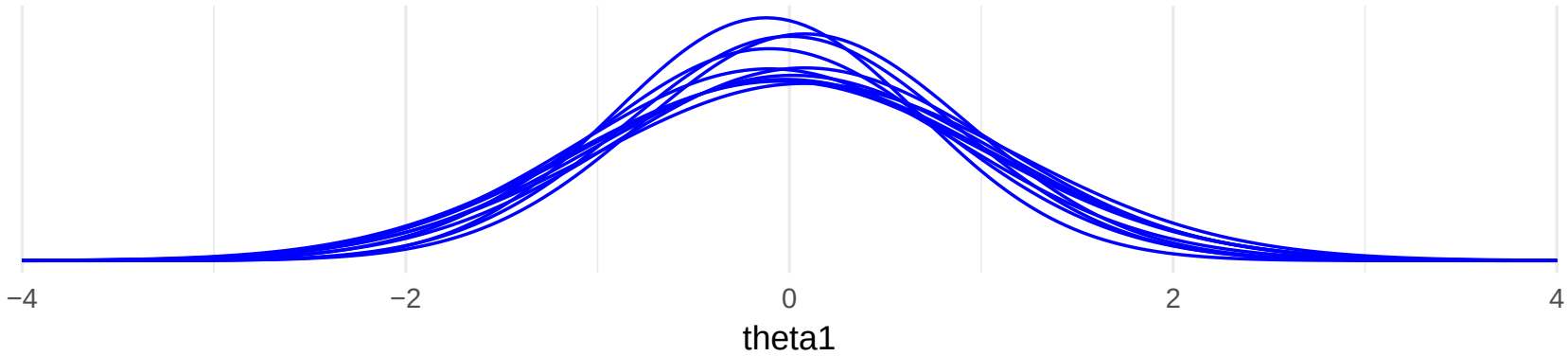
500 warmup, 500 post warmup iterations



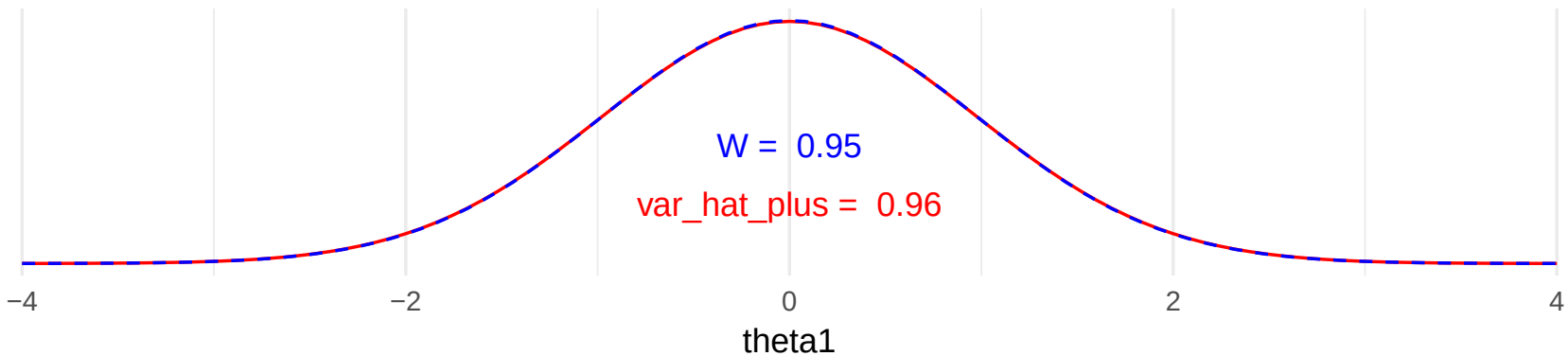
Rhat = 1.08



5000 warmup, 5000 post warmup iterations



Rhat = 1



- M chains, each having N draws
- Within chains variance W

$$W = \frac{1}{M}\sum_{m=1}^{M} s_m^2, \text{ where } s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta_{nm} - \bar{\theta}_{.m})^2$$

- Between chains variance B

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_{.m} - \bar{\theta}_{..})^2,$$

$$\text{where } \bar{\theta}_{.m} = \frac{1}{N}\sum_{n=1}^{N}\theta_{nm},\ \bar{\theta}_{..} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_{.m}$$

- Under the null (chains all converged), $E(W) = E(B) = \text{var}(\theta|y)$.
- So this leads to a weighted estimator of $\text{var}(\theta|y)$, which is unbiased if the chains converge,

$$\widehat{\text{var}}^+(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B$$

- $\widehat{\text{var}}^+$ overestimates marginal posterior variance in finite samples, since the chains are started from an overdispersed initial distribution.
- On the other hand, W *underestimates* marginal posterior variance.
  - Single chains have not yet visited all points in the distribution.
  - When $N \to \infty, \quad \text{E}(W) \to \text{var}(\theta|y)$.
- As $\widehat{\text{var}}^+(\theta|y)$ overestimates and W underestimates, compute

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^+}{W}}$$

- $\widehat{R} \to 1$ as $N \to \infty$
- If $\widehat{R}$ is big (e.g., $R > 1.01$), keep sampling.
- If $\widehat{R}$ close to 1, it is still possible that chains have not converged,
  - if starting points were not overdispersed
  - if distribution far from normal (especially if infinite variance)
  - just by chance when n is finite

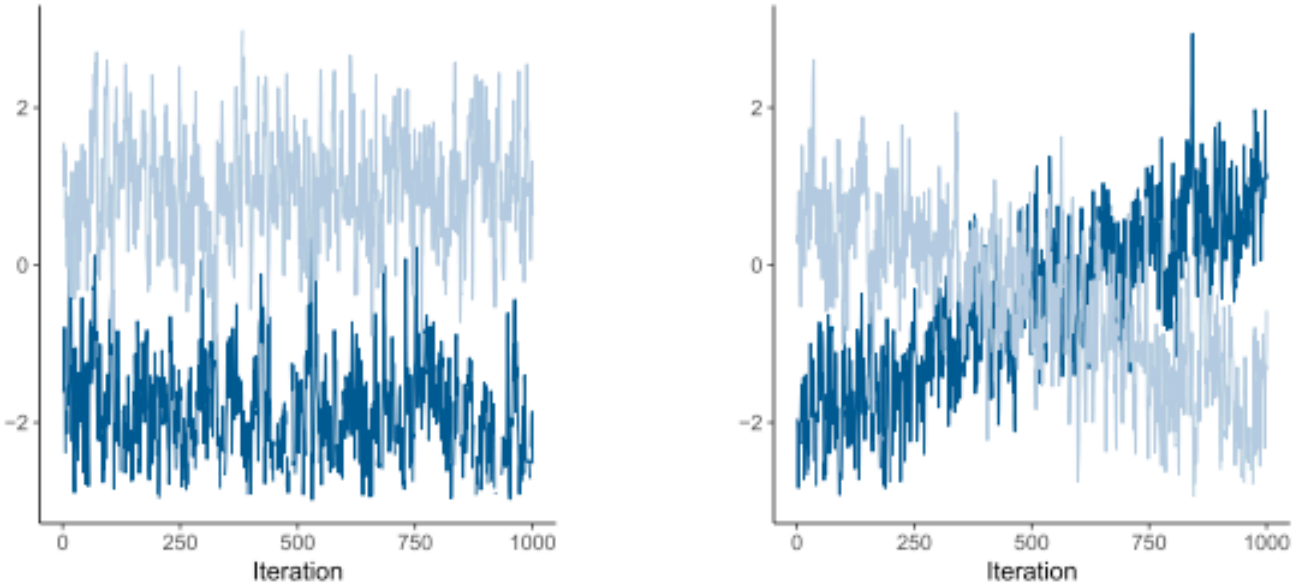


Figure 1: Examples of two challenges in assessing convergence of iterative simulations. (a) In the left plot, either sequence alone looks stable, but the juxtaposition makes it clear that they have not converged to a common distribution. (b) In the right plot, the two sequences happen to cover a common distribution but neither sequence appears stationary. These graphs demonstrate the need to use between-sequence and also within-sequence information when assessing convergence. Adapted from Gelman et al. (2013).

Split-$\widehat{R}$

- Split-$\widehat{R}$
  - To examine stationarity chains are split to two parts
  - after splitting, we have M chains, each having N draws
  - scalar draws $\theta_{nm} \quad (n = 1, \ldots, N; m = 1, \ldots, M)$
  - compare means and variances of the split chains
- Rank normalized $\widehat{R}$
  - Original $\widehat{R}$ requires that the target distribution has finite mean and variance.
  - Vehtari, Gelman, Simpson, Carpenter, Bürkner (2020). Rank-normalization, folding, and localization: An improved R-hat for assessing convergence of MCMC. Bayesian Analysis.